

# Programsko inženjerstvo

Ak. god. 2023./2024.

## Nestali ljubimci

Dokumentacija, Rev. 2

Grupa: *MP7*

Voditeljica: *Mia Krstičević*

Datum predaje: *19.01.2024.*

Nastavnik: *Alan Jović*

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2 Opis projektnog zadatka</b>	<b>7</b>
<b>3 Specifikacija programske potpore</b>	<b>12</b>
3.1 Funkcionalni zahtjevi . . . . .	12
3.1.1 Obrasci uporabe . . . . .	14
3.1.2 Sekvencijski dijagrami . . . . .	20
3.2 Ostali zahtjevi . . . . .	23
<b>4 Arhitektura i dizajn sustava</b>	<b>24</b>
4.1 Baza podataka . . . . .	25
4.1.1 Opis tablica . . . . .	25
4.1.2 Dijagram baze podataka . . . . .	29
4.2 Dijagram razreda . . . . .	30
4.3 Dijagram stanja . . . . .	31
4.4 Dijagram aktivnosti . . . . .	33
4.5 Dijagram komponenti . . . . .	34
<b>5 Implementacija i korisničko sučelje</b>	<b>35</b>
5.1 Korištene tehnologije i alati . . . . .	35
5.2 Ispitivanje programskog rješenja . . . . .	36
5.2.1 Ispitivanje komponenti . . . . .	36
5.2.2 Ispitivanje sustava . . . . .	39
5.3 Dijagram razmještaja . . . . .	49
5.4 Upute za puštanje u pogon . . . . .	50
5.4.1 Konfiguracija baze podataka . . . . .	50
5.4.2 Izgradnja aplikacije . . . . .	50
5.4.3 Deploy . . . . .	52
<b>6 Zaključak i budući rad</b>	<b>53</b>

<b>Popis literature</b>	<b>54</b>
<b>Indeks slika i dijagrama</b>	<b>56</b>
<b>Dodatak: Prikaz aktivnosti grupe</b>	<b>57</b>

# 1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljena shema baze podataka.	Mia Krstičević	27.10.2023.
0.2	Napravljen predložak projektne dokumentacije i dodan opis projektnog zadatka. Dopisane upute za povijest dokumentacije.	Lucija Renić	28.10.2023.
0.3	U specifikaciju programske potpore dodani funkcionalni zahtjevi. Dodani Ostali zahtjevi i Obrasci uporabe	Lucija Renić	30.10.2023.
0.4	Nova verzija baze podataka	Mia Krstičević	05.11.2023.
0.5	Uređen opis projektnog zadatka Dodan opis baze	Mia Krstičević	07.11.2023.
0.6.1	Dodan dijagram baze podataka	Mia Krstičević	09.11.2023.
0.6.2	Dodan UML dijagram obrasca uporabe	Toni Vanjak, Lucija Renić	09.11.2023.
0.7	Ispravljen UML dijagram obrasca uporabe	Toni Vanjak, Lucija Renić	11.11.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
0.8	Dodani sekvencijski dijagrami	Toni Vanjak, Lucija Renić	11.11.2023.
0.9	Dodani opisi sekvencijskih dijagrama	Lucija Renić	15.11.2023.
0.10	Dodan opis arhitekture	Toni Vanjak	16.11.2023.
0.11	Ispravljen sekvencijski dijagram UC7	Toni Vanjak	17.11.2023.
0.12	Dodan UC15 i popravljen dijagram obrazaca uporabe	Toni Vanjak, Mia Krstičević, Lucija Renić	17.11.2023.
0.13	Uređen opis projektnog zadatka	Mia Krstičević, Lucija Renić	17.11.2023.
0.14	Ispravljen sekvencijski dijagram UC3	Toni Vanjak	17.11.2023.
0.15	Dodan dijagram razreda	Toni Vanjak	17.11.2023.
1.1	Dodan dijagram stanja i njegov opis	Lucija Renić	10.01.2024.
1.2	Dodan dijagram aktivnosti i njegov opis	Lucija Renić	12.01.2024.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.3	Popravljen dijagram obrazaca uporabe	Lucija Renić, Toni Vanjak	15.01.2024.
1.4	Promijenjen popis obrazaca uporabe	Lucija Renić	15.01.2024.
1.5	Dodan dijagram razmještaja i njegov opis	Toni Vanjak, Lucija Renić	16.01.2024.
1.6	Popravljeni sekvencijski dijagrami	Toni Vanjak, Lucija Renić	16.01.2024.
1.7	U opis projektnog zadatka dodan odlomak "Postojeća rješenja"	Lucija Renić	16.01.2024.
1.8	Ispravljen dijagram razreda	Toni Vanjak	16.01.2024.
1.9	Dodan dijagram komponenti	Toni Vanjak	17.01.2024.
1.10	Započeto poglavlje "Ispitivanje programskog rješenja"	Lucija Renić	18.01.2024.
1.11	Završeno poglavlje "Ispitivanje programskog rješenja"	Lucija Renić	19.01.2024.
1.12	Uređen Dnevnik sastajanja i Tablica aktiv- nosti	Lucija Renić	19.01.2024.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.13	Dodan "Zaključak"	Lucija Renić	19.01.2024.
1.14	Dodan dijagram aktivnosti na repozitoriju	Lucija Renić	19.01.2024.
1.15	Dodano poglavlje "Upute za puštanje u po- gon"	Fran Kufrin	19.01.2024.

## 2. Opis projektnog zadatka

Cilj ovog projektnog zadatka je razviti web aplikaciju „Nestali ljubimci“ koja će korisniku olakšati potragu za odlutalim kućnim ljubimcem. Budući da je informacije o potrazi nepraktično oglašavati putem uobičajenih internetskih platformi za komunikaciju (npr. društvene mreže, stranice za oglašavanje, forumi i slično), ova web aplikacija će vlasnicima životinja, skloništima za životinje i dobrim ljudima koji pomažu u pronalasku životinje omogućiti brz i efikasan način izmjene informacija o nestanku, opažanju i pronalasku odbjeglog kućnog ljubimca. Radi što bržeg pronalaska ljubimca, web aplikacija će biti razvijena za mobilne uređaje kako bi u procesu potrage podržala veću brzinu reagiranja, koja je često ključna za uspješan završetak potrage za kućnim ljubimcem.

Prilikom ulaska u web aplikaciju prikazuju se aktivni oglasi, točnije nestali ljubimci. Oglase mogu postaviti uobičajeni korisnici ili skloništa. Poredak oglasa ovisi o načinu pretraživanja koji korisnik odabire.

### KORISNICI

Aplikacija podržava rad tri tipa korisnika (neregistrirani korisnik, registrirani korisnik i sklonište za životinje).

Neregistrirani korisnik pri ulasku u web aplikaciju ima mogućnost pregledavati i pretraživati aktivne oglase o nestalim kućnim ljubimcima. Pretraživanje je ostvareno po kategoriji oglasa i po kategorijama podataka o ljubimcu koje su dostupne pri oglašavanju (vrsta, ime na koje se odaziva, datum i sat nestanka, lokacija nestanka, boja, starost, tekstni opis). Za detaljniji pregled informacija o kućnom ljubimcu kao i za pregled komunikacije o potrazi za njime potrebno je odabrati oglas. Ako neregistrirani korisnik želi sudjelovati u komunikaciji oko potrage za kućnim ljubimcem, potrebna je registracija. Podaci potrebni za registraciju su:

- Ime
- Prezime
- Korisničko ime
- Lozinka



- Adresa e-pošte
- Broj telefona

The registration form is titled "Registracija". It contains several input fields: "Username", "Password", "Email", "Broj telefona", "Ime", and "Prezime". There is also a dropdown menu for "Vrsta korisnika" with "Korisnik" selected. A brown button labeled "Registriraj me" is at the bottom. Below the button is a link: "Već si registriran? [Prijavi se](#)".

Slika 2.1: Registracija

The login form is titled "Prijava". It contains two input fields: "Username" and "Password". A brown button labeled "Prijavi se" is at the bottom. Below the button is a link: "Nisi registriran? [Registriraj se](#)".

Slika 2.2: Prijava

Registrirani korisnik, koji osim svih mogućnosti koje ima i neregistrirani korisnik, dodatno ima sljedeće mogućnosti:

- Postavljanje oglasa o nestalom kućnom ljubimcu

- Sudjelovanje u komunikaciji oko potrage za ljubimcem
- Uklanjanje oglasa o nestalom kućnom ljubimcu
- Izmjena oglasa o nestalom kućnom ljubimcu

Treći tip korisnika su skloništa za životinje, koji osim svih mogućnosti koje ima registrirani korisnik, ima dodatnu mogućnost oglašavanja životinja koje su pronađene i nalaze se u njihovom prostoru. Skloništa za životinje pri registraciji moraju unijeti i naziv skloništa.

## OGLASI

Za postavljanje oglasa potrebno je unijeti sljedeće informacije o ljubimcu:

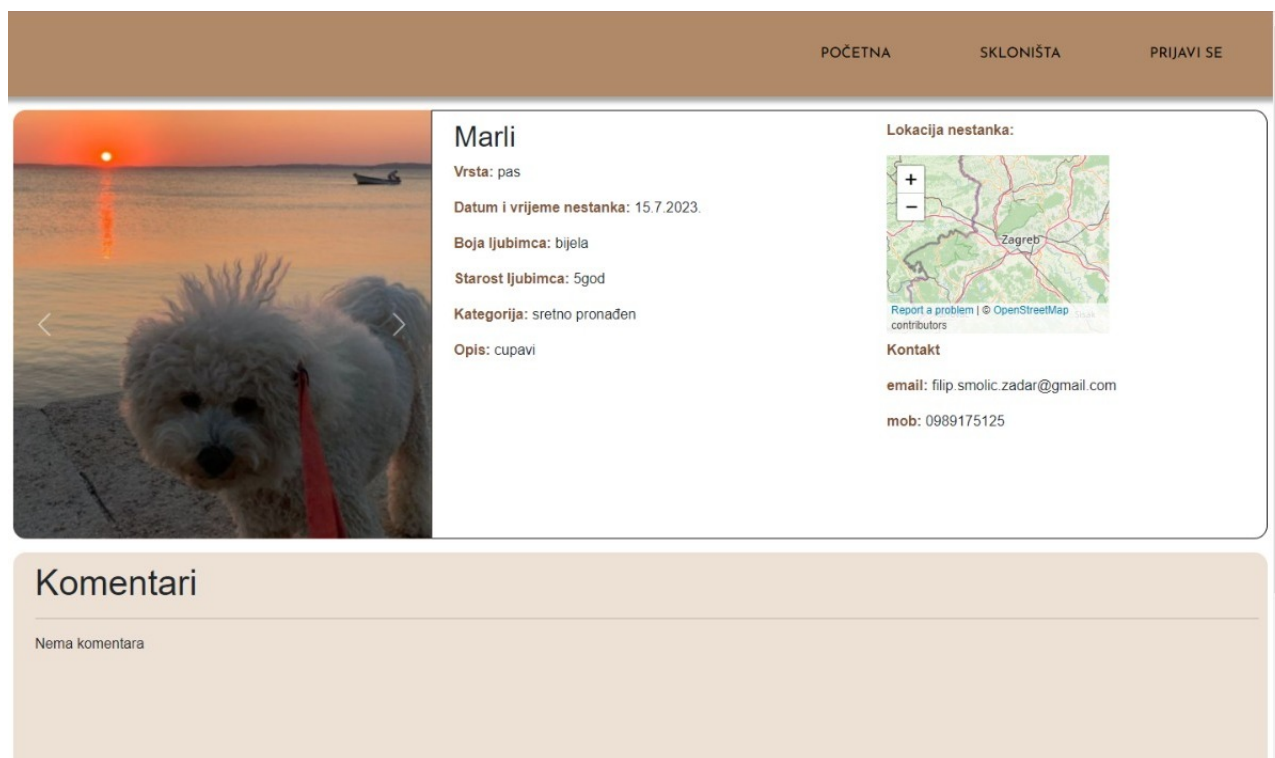
- Vrsta
- Ime na koje se odaziva
- Datum i sat nestanka
- Lokacija nestanka
- Boja
- Starost
- Tekstni opis
- Slike (najviše 3 slike)

Pri pretraživanju kućnih ljubimaca, vrsta je podijeljena u kategorije: pas, mačka, ptica, glodavac, kunić, gmaz te ostalo.

Uz to, postoje kategorije za boju (crna, smeđa, bijela, siva, zelena, crvena, žuta, narančasta, plava, šarena) i za starost. Starost je podijeljena u intervale: <1god., 1god., 2god., 3god., 4-5god., 6-10god. i >10god. Osim podataka o kućnom ljubimcu, oglas sadrži i kontakt podatke korisnika koji se povlače iz korisničkih podataka danih pri registraciji (adresa e-pošte, broj telefona).

Oglas mora imati jednu od kategorija:

- Za ljubimcem se traga
- Ljubimac je sretno pronađen
- Ljubimac nije pronađen, ali se za njim više ne traga aktivno
- Ljubimac je pronađen uz nesretne okolnosti
- U skloništu



Slika 2.3: Primjer oglasa

Od ovih kategorija inicijalno je postavljena da se za ljubimcem traga. Kategoriju 'U skloništu' imaju mogućnost postaviti samo prijavljena skloništa za životinje. Na oglasu je moguća izmjena svih kategorija podataka o ljubimcima, kao i kategorija oglasa. Sve kategorije, osim da se za ljubimcem aktivno traga, oglas čine neaktivnim. Također, web aplikacija sadrži popis neaktivnih oglasa koje mogu pregledavati samo registrirani korisnici.

Komunikacija oko potrage za kućnim ljubimcem odvijat će se porukama koje mogu sadržavati:

- Tekst
- Sliku
- Geolokaciju

uz kontakt podatke o osobi koja komunicira. Geolokacija je izvedena uz pomoć vanjske usluge OpenStreetMap.

### Ostalo

Aplikacija je izvedena kao mobilna aplikacija.

Sustav podržava rad više korisnika u stvarnom vremenu. Također, sustav podržava

hrvatski jezik.

Aplikacija je namijenjena vlasnicima izgubljenih ljubimaca i svim dobrim dušama koje su voljne pomoći oko potrage za nestalim kućnim ljubimcima.

### **Postojeća rješenja**

Potrage oko nestalih kućnih ljubimaca trenutno se odvijaju putem društvenih mreža, na primjer facebook grupa za nestale i nađene ljubimce, putem instagram profila, njuškala i web stranica azila.

### **Moguća buduća proširenja aplikacije:**

- obavješćavanje vlasnika izgubljenog kućnog ljubimca nakon što korisnik ostavi komentar na njegovom oglasu
- integracija s postojećim društvenim mrežama

## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

#### Dionici:

1. Vlasnik (naručitelj)
2. Vlasnici kućnih ljubimaca
3. Ljudi koji pomažu u potrazi
4. Skloništa za životinje
5. Razvojni tim

#### Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:
  - (a) registrirati se, stvoriti novi korisnički račun za koje je potrebno unijeti ime, prezime, korisničko ime, lozinku, adresu e-pošte, broj telefona
  - (b) pregledavati i pretraživati aktivne oglase
  - (c) odabrati oglas za detaljniji pregled informacija o kućnom ljubimcu i pregled komunikacije oko kućnog ljubimca
2. Vlasnik kućnog ljubimca (inicijator) može:
  - (a) izbrisati korisnički račun
  - (b) pregledavati i mijenjati osobne podatke
  - (c) pregledavati aktivne i neaktivne oglase
  - (d) postaviti oglas o nestalom kućnom ljubimcu za koji je potrebno unijeti vrstu, ime na koje se odaziva, datum i sat nestanka, lokaciju nestanka, boju, starost, tekstni opis i slike (do 3)
  - (e) sudjelovati u komunikaciji o nestalom kućnom ljubimcu
    - i. sudjelovati slanjem tekstne poruke
    - ii. sudjelovati slanjem slike
    - iii. sudjelovati slanjem geolokacije
  - (f) ukloniti oglas o nestalom kućnom ljubimcu

(g) izmjenjivati oglas o nestalom kućnom ljubimcu

3. Sudionik u potrazi (inicijator) može:

- (a) izbrisati korisnički račun
- (b) pregledavati i mijenjati osobne podatke
- (c) pregledavati aktivne i neaktivne oglase
- (d) sudjelovati u komunikaciji o nestalom kućnom ljubimcu
  - i. sudjelovati slanjem tekstne poruke
  - ii. sudjelovati slanjem slike
  - iii. sudjelovati slanjem geolokacije

4. Sklonište za životinje (inicijator) može:

- (a) izbrisati korisnički račun
- (b) pregledavati aktivne i neaktivne oglase
- (c) postaviti oglas o pronađenoj životinji
- (d) sudjelovati u komunikaciji o nestalom kućnom ljubimcu
  - i. sudjelovati slanjem tekstne poruke
  - ii. sudjelovati slanjem slike
  - iii. sudjelovati slanjem geolokacije

5. Baza podataka (sudionik) može:

- (a) pohraniti podatke o korisnicima
- (b) pohraniti podatke o kućnim ljubimcima (oglas)

### 3.1.1 Obrasci uporabe

#### UC1 - Pregled aktivnih oglasa

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati oglase o nestalim kućnim ljubimcima
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Aktivni oglasi su prikazani prilikom učitavanja aplikacije
  2. Korisnik odabire oglas za više informacija
  3. Prikazuju se informacije i komunikacija o nestalom kućnom ljubimcu

#### UC2 - Registracija

- **Glavni sudionik:** Korisnik
- **Cilj:** Stvoriti korisnički račun za pristup sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za registraciju
  2. Korisnik unosi potrebne korisničke podatke
  3. Korisnik prima obavijest o uspješnoj registraciji
- **Opis mogućih odstupanja:**
  - 2.a Odabir već zauzetog korisničkog imena i/ili e-maila, unos korisničkog podatka u nedozvoljenom formatu ili unos neispravnoga e-maila
    1. Sustav obavještava korisnika o pogrešnom unosu podataka i vraća ga na stranicu za registraciju
    2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije

#### UC3 - Prijava u sustav

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Dobiti pristup korisničkom sučelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
  1. Unos korisničkog imena i lozinke

- 2. Potvrda o ispravnosti unesenih podataka
- 3. Pristup korisničkim funkcijama
- **Opis mogućih odstupanja:**
  - 2.a Neispravno korisničko ime/lozinka
    - 1. Sustav obavještava korisnika o neuspjeloj prijavi i vraća ga na stranicu za prijavu u sustav
    - 2. Korisnik unosi ispravne podatke i uspješno se prijavljuje u sustav

#### UC4 - Postavljanje oglasa

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Postavljanje oglasa o nestalom kućnom ljubimcu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  - 1. Korisnik odabire opciju "Novi oglas"
  - 2. Korisnik unosi podatke o nestalom kućnom ljubimcu
  - 3. Korisnik odabire opciju "Postavi oglas"
- **Opis mogućih odstupanja:**
  - 3.a Korisnik ne unosi obvezne podatke o ljubimcu
    - 1. Sustav obavještava korisnika da nisu uneseni svi potrebni podaci
    - 2. Korisnik unosi sve podatke
    - 3. Korisnik postavlja oglas
  - 3.b Korisnik odustaje od postavljanja oglasa
    - 1. Sustav obavještava korisnika da oglas neće biti postavljen
    - 2. Prikazuje se početna stranica

#### UC5 - Pregled mojih oglasa

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Pregledati listu oglasa koje je postavio korisnik
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  - 1. Korisnik odabire opciju "Moji oglasi"
  - 2. Aplikacija prikazuje listu oglasa koje je postavio korisnik
- **Opis mogućih odstupanja:**
  - 2.a Korisnik nema postavljene oglase



1. Sustav obavještava korisnika da još nije postavio oglas

#### UC6 - Izmjena kategorije oglasa

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Promijeniti kategoriju oglasa
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju "Moji oglasi"
  2. Odabire oglas kojem želi promijeniti kategoriju
  3. Odabire opciju "Uredi"
  4. Korisnik odabire jednu od ponuđenih kategorija
  5. Korisnik odabire opciju "Spremi"
  6. Oglas je promijenjen
- **Opis mogućih odstupanja:**
  - 2.a UC8
  - 5.a Korisnik ne odabire opciju "Spremi"
    1. Oglas ostaje nepromijenjen

#### UC7 - Uklanjanje oglasa

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Ukloniti oglas
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju "Moji oglasi"
  2. Odabire oglas kojeg želi izbrisati
  3. Korisnik odabire opciju "Izbriši"
- **Opis mogućih odstupanja:**
  - 3.a korisnik odustaje od brisanja oglasa
    1. Korisnik odabire opciju "Odustani"
    2. Aplikacija prikazuje listu korisnikovih oglasa

#### UC8 - Komunikacija

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Komunikacija oko pronalaska kućnog ljubimca

- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire oglas o kućnom ljubimcu
  2. Korisnik odabire opciju "Poruke"
  3. Aplikacija prikazuje poruke vezane za komunikaciju oko potrage za kućnim ljubimcem

#### UC9 - Slanje poruke

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Poslati poruku s informacijama o kućnom ljubimcu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. UC11
  2. Korisnik odabire opciju "Pošalji poruku"
  3. Korisnik unosi poruku
  4. Korisnik odabire opciju "Pošalji" i šalje poruku
- **Opis mogućih odstupanja:**
  - 3.a korisnik odustaje od slanja poruke
    1. Korisnik odabire opciju "Odustani"
    2. Aplikacija prikazuje poruke vezane za komunikaciju oko potrage za kućnim ljubimcem

#### UC10 - Slanje slike

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Poslati sliku u poruci
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. UC11
  2. Korisnik odabire opciju "Pošalji poruku"
  3. Korisnik odabire opciju "Dodaj sliku u privitak"
  4. Korisnik odabire sliku
  5. Korisnik odabire opciju "Pošalji" i šalje poruku
- **Opis mogućih odstupanja:**

4.a UC12 3.a

5.a UC12 3.a

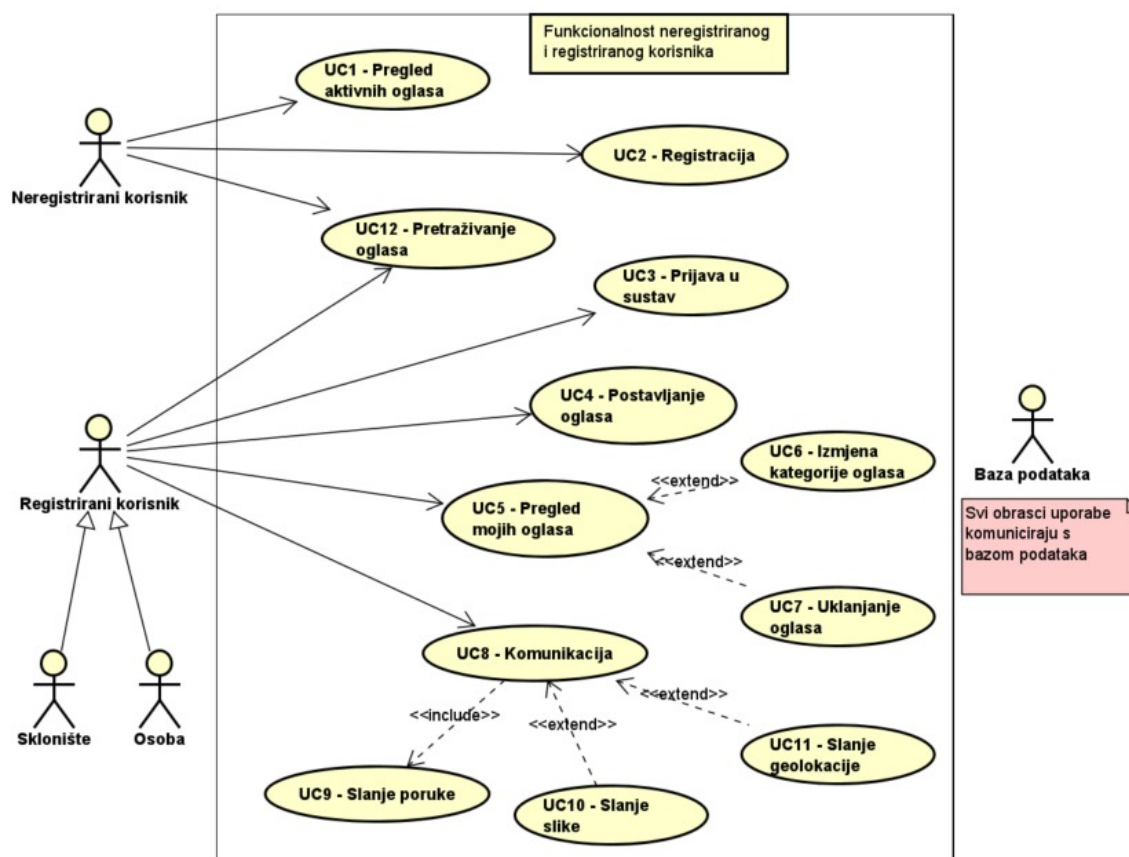
### UC11 - Slanje geolokacije

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Poslati geolokaciju u poruci
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. UC11
  2. Korisnik odabire opciju "Pošalji poruku"
  3. Korisnik odabire opciju "Dodaj geolokaciju"
  4. Korisnik odabire geolokaciju
  5. Korisnik odabire opciju "Pošalji" i šalje poruku
- **Opis mogućih odstupanja:**
  - 4.a UC12 3.a
  - 5.a UC12 3.a

### UC12 - Pretraživanje oglasa

- **Glavni sudionik:** Korisnik
- **Cilj:** Pretražiti oglas po kategoriji
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire kategoriju po kojoj želi pretraživati (jednu ili više)
  2. Korisnik odabire opciju "Pretraži"
  3. Aplikacija prikazuje filtrirane oglase
- **Opis mogućih odstupanja:**
  - 1.a Korisnik unosi nevaljani datum, grad ili ulicu
    1. Sustav javlja korisniku da je unesen nevaljani podatak za pretragu

## Dijagrami obrazaca uporabe

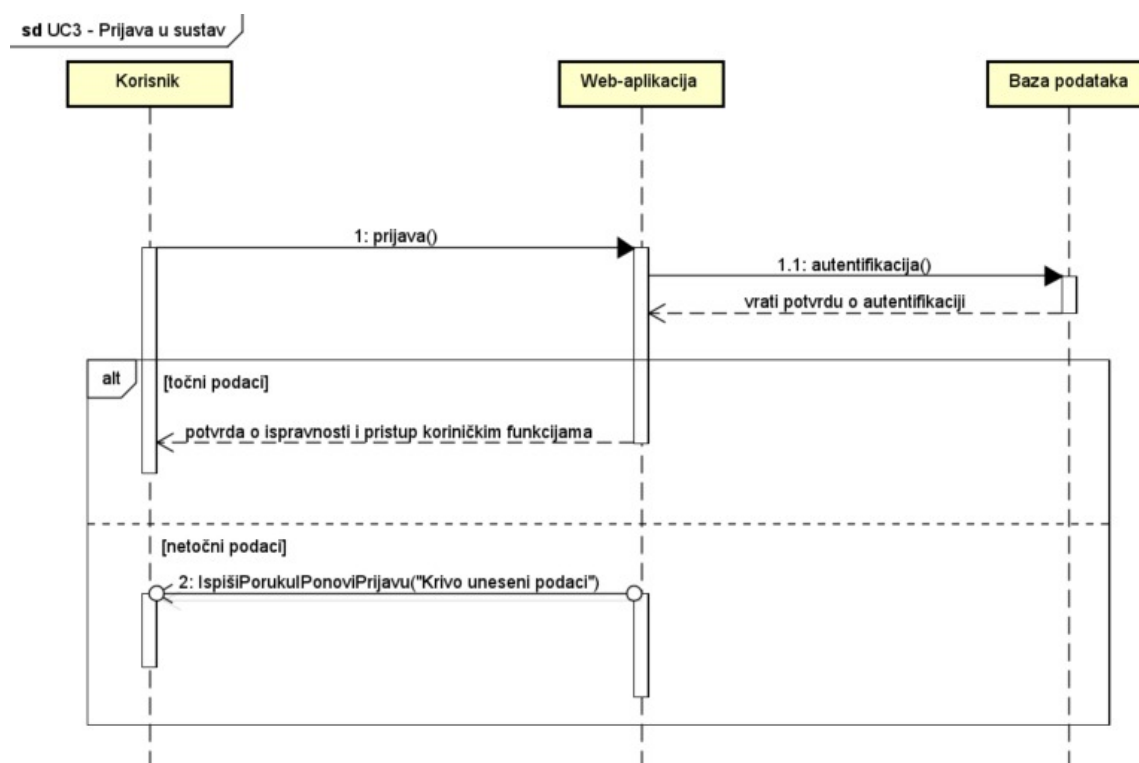


Slika 3.1: Dijagram obrasca uporabe, funkcionalnosti neregistriranog i registriranog korisnika

### 3.1.2 Sekvencijski dijagrami

#### Obrazac uporabe UC3 - Prijava u sustav

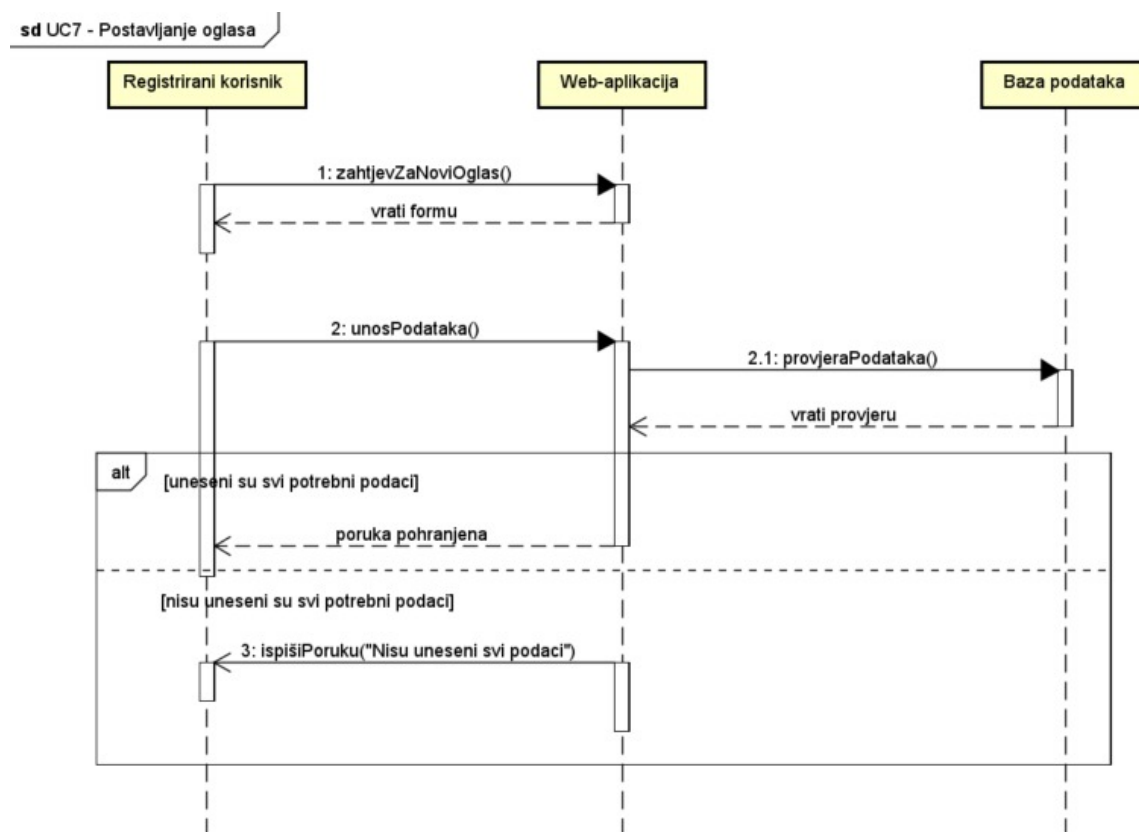
Korisnik odabire opciju "Prijava se". Sustav prikazuje formu za prijavu, te korisnik unosi potrebne podatke. Sustav provjerava postoje li uneseni podaci u bazi podataka. Ako su uneseni podaci ispravni, sustav obavještava korisnika da je prijava uspjela te mu omogućava pristup korisničkim funkcijama. U slučaju unosa pogrešnih podataka sustav obavještava korisnika o pogrešci, te se ponovno prikazuje forma za prijavu.



Slika 3.2: Sekvencijski dijagram za UC3

**Obrazac uporabe UC7 - Postavljanje oglasa**

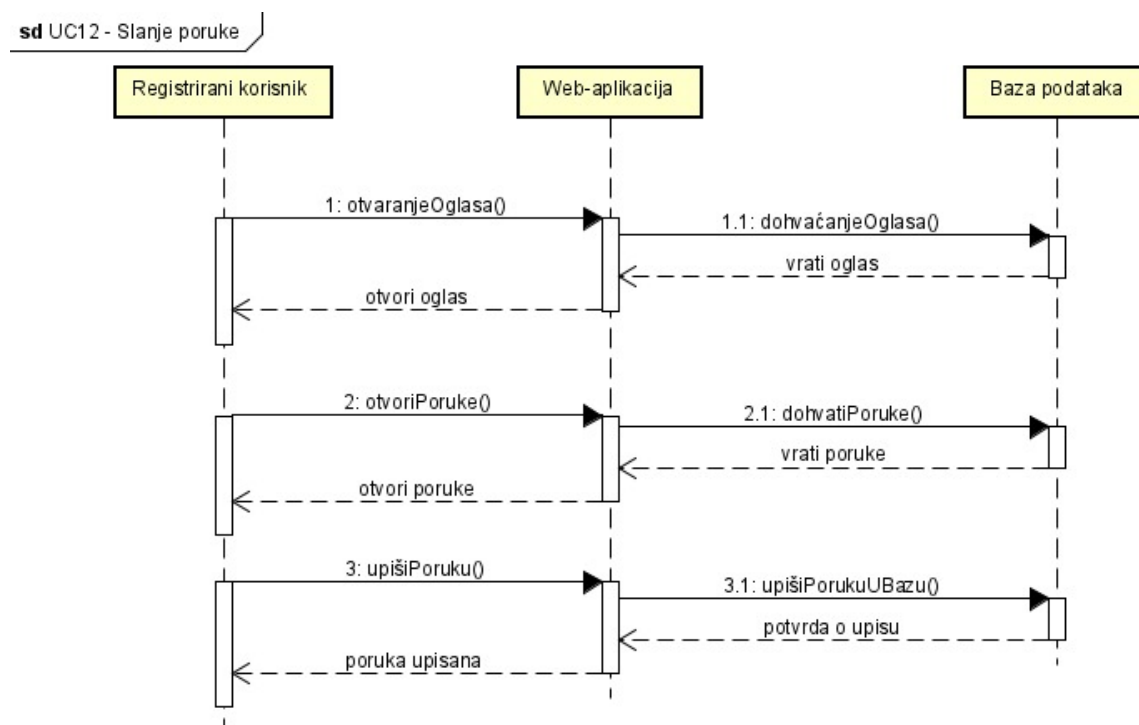
Korisnik odabire opciju "Novi oglas". Sustav prikazuje formu za unos podataka o izgubljenom kućnom ljubimcu. Korisnik unosi podatke, te odabire opciju "Spremi". Sustav provjerava u bazi podataka jesu li uneseni svi potrebni podaci. Baza podataka vraća provjeru sustavu, te ako su uneseni svi potrebni podaci za postavljanje oglasa, oglas se pohranjuje u bazu podataka i sustav obavještava korisnika da je oglas uspješno spremljen. U suprotnom, sustav obavještava korisnika da nisu uneseni svi potrebni podaci.



Slika 3.3: Sekvencijski dijagram za UC7

### Obrazac uporabe UC12 - Slanje poruke

Kako bi korisnik poslao poruku, najprije mora odabrati oglas na kojem želi ostaviti poruku. Nakon odabira oglasa iz baze podataka se dohvaćaju podaci vezani za taj oglas. Sustav prikazuje oglas (podatke o nestalom kućnom ljubimcu). Korisnik odabire opciju "Prikaži poruke", te se iz baze podataka dohvaćaju sve dosadašnje poruke vezane za nestalog kućnog ljubimca. Sustav sada, uz podatke o ljubimcu, prikazuje i svu dosadašnju komunikaciju vezanu za nestalog kućnog ljubimca. Korisnik odabire opciju "Nova poruka" i unosi željeni tekst, te opcionalno sliku ili geolokaciju. Nakon što korisnik odabere opciju "Pošalji poruku", poruka se upisuje u bazu podataka. Nakon što je poruka uspješno upisana u bazu podataka, sustav obavještava korisnika da je poruka poslana.



Slika 3.4: Sekvencijski dijagram za UC12

## 3.2 Ostali zahtjevi

- Sustav treba omogućiti rad više korisnika u stvarnom vremenu
- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost sustava
- Aplikacija treba biti izvedena kao web aplikacija prilagođena za mobilne uređaje (responzivna)
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu (dijakritičke znakove) pri unosu i prikazu tekstualnog sadržaja
- Sustav treba biti jednostavan za korištenje



## 4. Arhitektura i dizajn sustava

Arhitektura se dijeli na tri podsustava:

- web poslužitelj
- web aplikaciju
- bazu podataka

Web aplikacije često zahtijevaju integraciju poslužitelja, aplikacije i baze podataka kako bi pružile korisnicima dinamičke i interaktivne sadržaje. **Web poslužitelj** služi kao središnja točka koja prima zahtjeve korisnika putem internetskog preglednika. **Web aplikacija** obrađuje zahtjev te ovisno o njemu komunicira s bazom podataka kako bi dohvatila, mijenjala ili pohranjivala podatke koji se koriste u procesu. **Baza podataka** je skup strukturiranih podataka koji se čuvaju na poslužitelju, a aplikacija je odgovorna za upravljanje tim podacima i osiguravanje njihove konzistentnosti. Kada se podaci mijenjaju putem aplikacije, te promjene se ažuriraju u bazi podataka, a zatim web poslužitelj šalje ažurirane informacije korisnicima putem preglednika.

Jezici korišteni pri izradi aplikacije su C#, CSS, HTML i JavaScript.

Arhitektura sustava se bazira na **MVC** konceptu. MVC (Model-View-Controller) je arhitekturni obrazac za razvoj softverskih aplikacija. **Model** predstavlja podatke i logiku aplikacije, **View** predstavlja sučelje preko kojeg korisnici komuniciraju s aplikacijom i prikazuje im podatke iz Modela, dok **Controller** upravlja komunikacijom između Modela i View-a. Ovaj koncept omogućava jasnu organizaciju koda, razdvajanje odgovornosti između komponenti aplikacije (Model, View, Controller) te olakšava timsku suradnju i održavanje aplikacije. Zbog jasne podjele na modele, poglede i kontrolere, aplikacije izgrađene na MVC arhitekturi često su fleksibilnije i lakše za održavanje.

## 4.1 Baza podataka

Za pohranu podataka iz naše web aplikacije napravljena je relacijska baza podataka. Tablicama i njihovim atributima ostvarena je komunikacija između organiziranosti, preciznosti i jednostavnog dohvaćanja podataka za daljnu obradu. Baza podataka je kreirana u SQLite-u, upravo zbog toga što se podaci iz baze mogu dijeliti između više računala i jer koristi jednu datoteku za pohranu cijele baze podataka.

Baza podataka se sastoji od sljedećih entiteta:

- User
- Regular
- Shelter
- TypeOfUser
- Communication
- Ad
- PhotoAd
- Pet
- ColorPet
- hasColor

### 4.1.1 Opis tablica

**Korisnik (User)** Ovaj entitet sadržava sve primarne informacije o korisniku. Sadrži attribute: userID, userName, email, phoneNum i psw. Ovaj entitet u vezi je *One-to-Many* s entitetom Ad (oglas), *One-to-Many* s entitetom Communication (Komunikacija) oba preko atributa userID korisnika te u vezi *One-to-One* s entitetom Regular (redovni korisnik), *One-to-One* s Shelter(sklonište) i *One-to-One* s TypeOfUser (tip korisnika) preko userID - korisničkog imena.

User		
userID	INTEGER	jedinstveni identifikator korisnika
userName	VARCHAR	naziv korisnika u aplikaciji
email	VARCHAR	e-mail adresa korisnika

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

User		
phoneNum	VARCHAR	broj telefona korisnika
psw	VARCHAR	lozinka korisnika

**Redovni korisnik (Regular)** Ovaj entitet sadržava informacije o regularnom korisniku. Sadrži attribute: userID, firstName - ime i lastName - prezime korisnika. Ovaj entitet u vezi je *One-to-One* s entitetom User (Korisnik) preko userID korisnika.

Regular		
userID	INTEGER	jedinstveni identifikator korisnika, (user.userID)
firstName	VARCHAR	ime korisnika
lastName	VARCHAR	prezime korisnika

**Sklonište (Shelter)** Ovaj entitet sadržava informacije o skloništu kao korisniku. Sadrži attribute: userID i nameShelter - naziv skloništa. Ovaj entitet u vezi je *One-to-One* s entitetom User (Korisnik) preko userID korisnika.

Shelter		
userID	INTEGER	jedinstveni identifikator korisnika, (user.userID)
nameShelter	VARCHAR	naziv skloništa za životinje

**Tip korisnika (TypeOfUser)** Ovaj entitet sadržava informacije o tipu korisnika. Korisnik može biti regular - uobičajen korisnik ili shelter - sklonište za životinje. Sadrži attribute: userID i userType - tip korisnika. Ovaj entitet u vezi je *One-to-One* s entitetom User (Korisnik) preko userID korisnika.

TypeOfUser		
userID	INTEGER	jedinstveni identifikator korisnika, (user.userID)
userType	VARCHAR	tip korisnika - može biti shelter ili regular

**Komunikacija (Communication)** Ovaj entitet sadržava sve važne informacije o komunikaciji ispod aktivnog oglasa. Sadrži attribute: textID, textCom - poruka, locCom - lokacija poruke te photoCom - slika unutar poruke te adID i userID - strani ključevi tablice Ad i User. Ovaj entitet u vezi je *Many-to-One* s entitetom Ad (oglas) preko atributa adID oglasa i *Many-to-One* s User preko userID korisnika.

Communication		
textID	INTEGER	jedinstveni identifikator poruke
photoCom	VARCHAR	slika poruke
textCom	VARCHAR	tekst poruke
locCom	VARCHAR	lokacija poruke
adID	INTEGER	jedinstveni identifikator oglasa, (ad.adID)
userID	INTEGER	jedinstveni identifikator korisnika, (user.userID)

**Oglas (Ad)** Ovaj entitet sadržava sve važne informacije o postavljanju oglasa. Sadrži attribute: adID, catID - kategorija oglasa, userID - strani ključ tablice User. Ovaj entitet u vezi je *Many-to-One* s entitetom User (korisnik) preko atributa userID korisnika, *One-to-Many* s Communication (Komunikacija) preko atributa adID, *One-to-Many* s PhotoAd (slike oglasa) preko adID, *One-to-One* s entitetom Pet(Ljubimac) preko adID oglasa.

Ad		
adID	INTEGER	jedinstveni identifikator oglasa
catAd	VARCHAR	kategorije oglasa

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Ad		
userID	INTEGER	jedinstveni identifikator korisnika, (user.userID)

**Slike oglas (PhotoAd)** Ovaj entitet sadržava sve važne informacije o postavljaju slika ispod oglasa. Sadrži attribute: photoID, photo - fotografija izgubljenog ljubimca, userID - strani ključ tablice User. Ovaj entitet u vezi je *Many-to-One* s entitetom Ad (Oglas) preko atributa adID. Uz to, dodano je i ograničenje na razini baze - trigger za postavljanje maksimalno 3 slika uz oglas.

PhotoAd		
photoID	INTEGER	jedinstveni identifikator slike oglasa
photo	VARCHAR	fotografija ljubimca
adID	INTEGER	jedinstveni identifikator oglasa, (ad.adID)

**Ljubimac (Pet)** Ovaj entitet sadržava sve važne informacije o nestalom kućnom ljubimcu. Sadrži attribute: petID, namePet - ime ljubimca, dateHourMis - datum i sat nestanka, location - lokacija pri postavljanju oglasa, vrsta ljubimca species, starost age, tekstni opis description i strani ključ adID iz Ad. Ovaj entitet u vezi je *One-to-One* s entitetom Ad (oglas) preko atributa adID, *Many-to-Many* s ColorPet (Komunikacija) preko tablice Has.

Pet		
petID	INTEGER	jedinstveni identifikator ljubimca
namePet	VARCHAR	ime na koje se odaziva ljubimac
dateHourMis	DATETIME	datum i sat nestanka ljubimca
location	VARCHAR	geolokacija izgubljenog ljubimca
species	VARCHAR	vrsta ljubimca
age	VARCHAR	starost ljubimca
description	VARCHAR	opis ljubimca

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Pet		
adID	INTEGER	jedinstveni identifikator oglasa, (ad.adID)

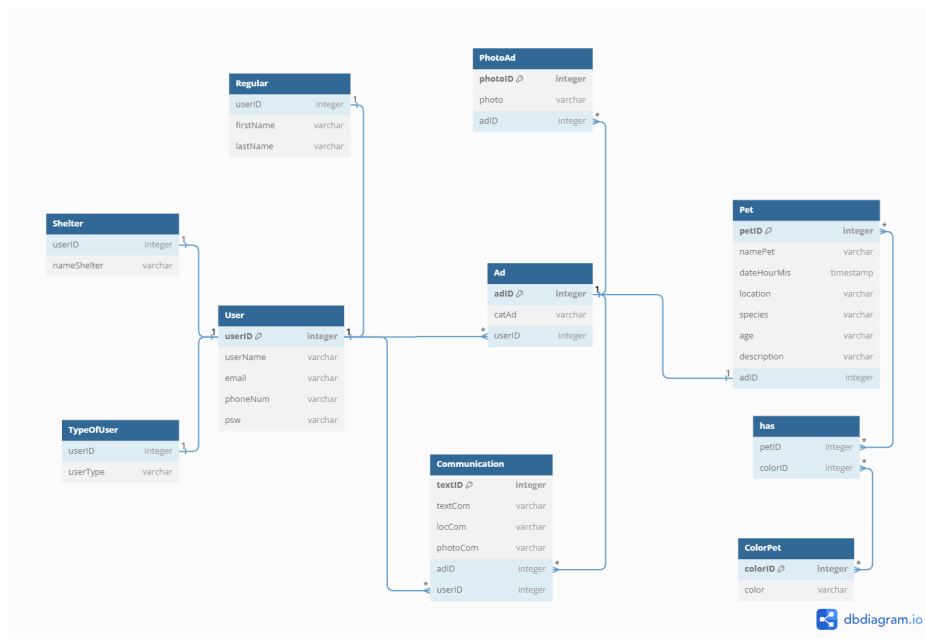
**Boja ljubimca (ColorPet)** Ovaj entitet sadržava sve važne informacije o boji kućnog ljubimca. Sadrži attribute: colorID i boja ljubimca color. Ovaj entitet u vezi je *Many-to-Many* s entitetom Pet(Ljubimac) preko tablice Has.

ColorPet		
colorID	INTEGER	jedinstveni identifikator boje ljubimca
color	VARCHAR	boja ljubimca

**ImaBoju (Has)** Ovaj entitet spaja boju i kućnog ljubimca. Sadrži attribute jedinstvene šifre colorID i petID. Ovaj entitet u vezi je *Many-to-Many* s entitetom Pet(Ljubimac) preko atributa petID te u vezi *Many-to-Many* s entitetom ColorPet(boja ljubimca) preko atributa colorID.

Has		
petID	INTEGER	jedinstveni identifikator ljubimca, (pet.petID)
colorID	INTEGER	jedinstveni identifikator boje ljubimca, (colorPet.colorID)

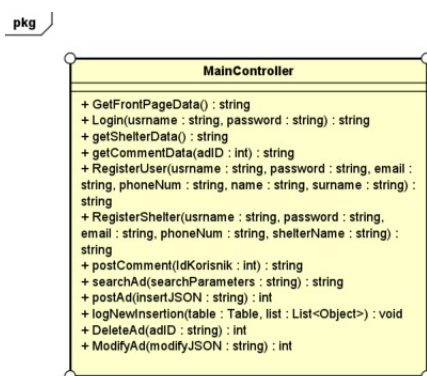
#### 4.1.2 Dijagram baze podataka



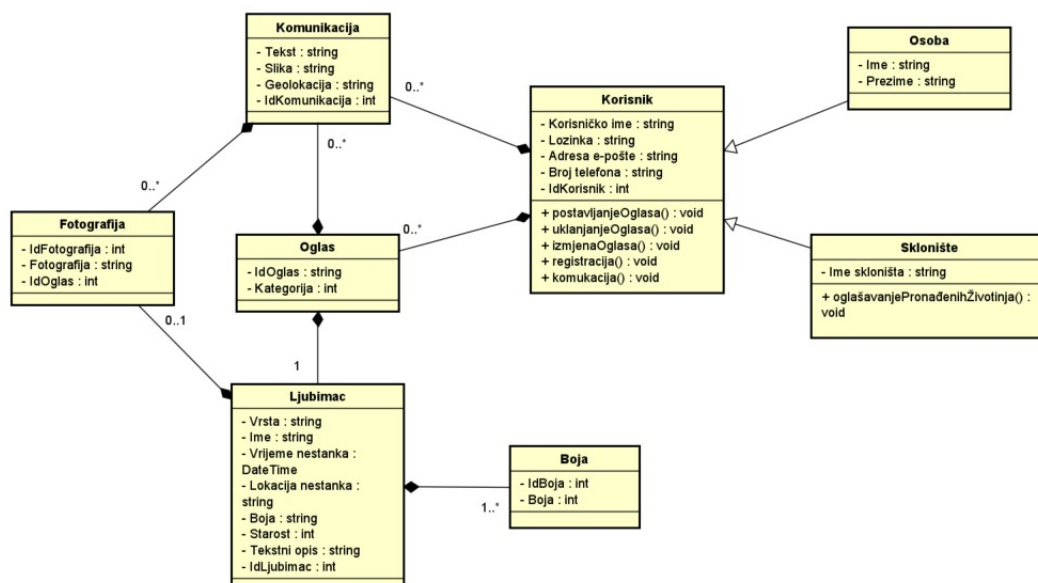
Slika 4.1: ER dijagram baze podataka

## 4.2 Dijagram razreda

Razredi modela odražavaju strukturu podataka unutar aplikacije, a metode koje su implementirane unutar njih služe za izravnu komunikaciju s bazom podataka kako bi dobili ili manipulirali traženim podacima. Razred Klijent predstavlja registri-ranog korisnika sustava s mogućnošću pristupa osnovnim funkcionalnostima sus-tava. Razred Sklonište ima sve funkcionalnosti kao Klijent, uz dodatnu mogućnost dodavanja kategorija oglasa. Razredi Ljubimac i Oglas sadrže informacije potrebne za stvaranje oglasa koji će biti prikazani svim korisnicima.



Slika 4.2: Dijagram razreda - dio Controllers

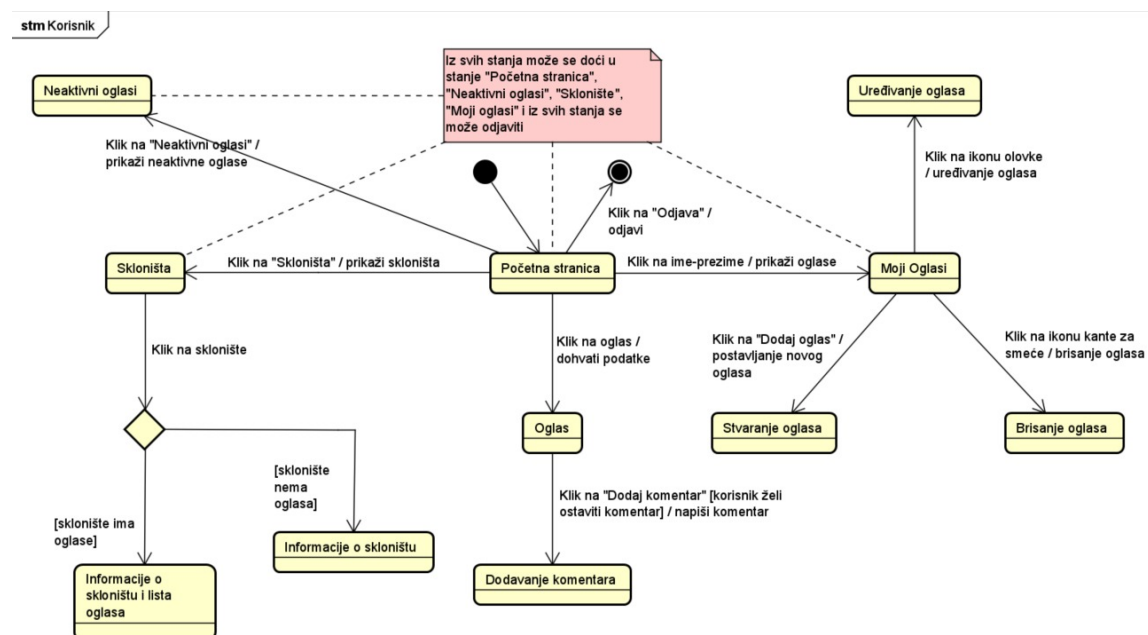


Slika 4.3: Dijagram razreda - dio Models

### 4.3 Dijagram stanja

Dijagram stanja strukturirano prikazuje kako sustav ili njegovi dijelovi prelaze iz jednog stanja u drugo kao odgovor na neki događaj. Na slici 4.4 prikazan je dijagram stanja za registriranog korisnika. Nakon prijave u sustav, korisniku se prikazuje početna stranica na kojoj je vidljiva lista oglasa. Odabirom oglasa, prikazuju se podaci o izgubljenom kućnom ljubimcu. Korisnik ima mogućnost ostaviti komentar na oglasu ukoliko ima informaciju koja bi mogla pomoći u potrazi za izgubljenom životinjom. Korisnik može odabrati neku od mogućnosti iz navigacijske trake. Odabirom na „Neaktivni oglasi“, prikazuju se oglasi svih kategorija, osim one da se za ljubimcem aktivno traga. Klikom na „Skloništa“, prikazuje se lista skloništa, a odabirom na neko od skloništa prikazuju se podaci o skloništu te oglasi od odabranog skloništa, ako ih ima. Klikom na korisničko ime, prikazuju se oglasi korisnika, koje korisnik može urediti ili izbrisati. Također, korisnik ima mogućnost stvaranja novog oglasa.

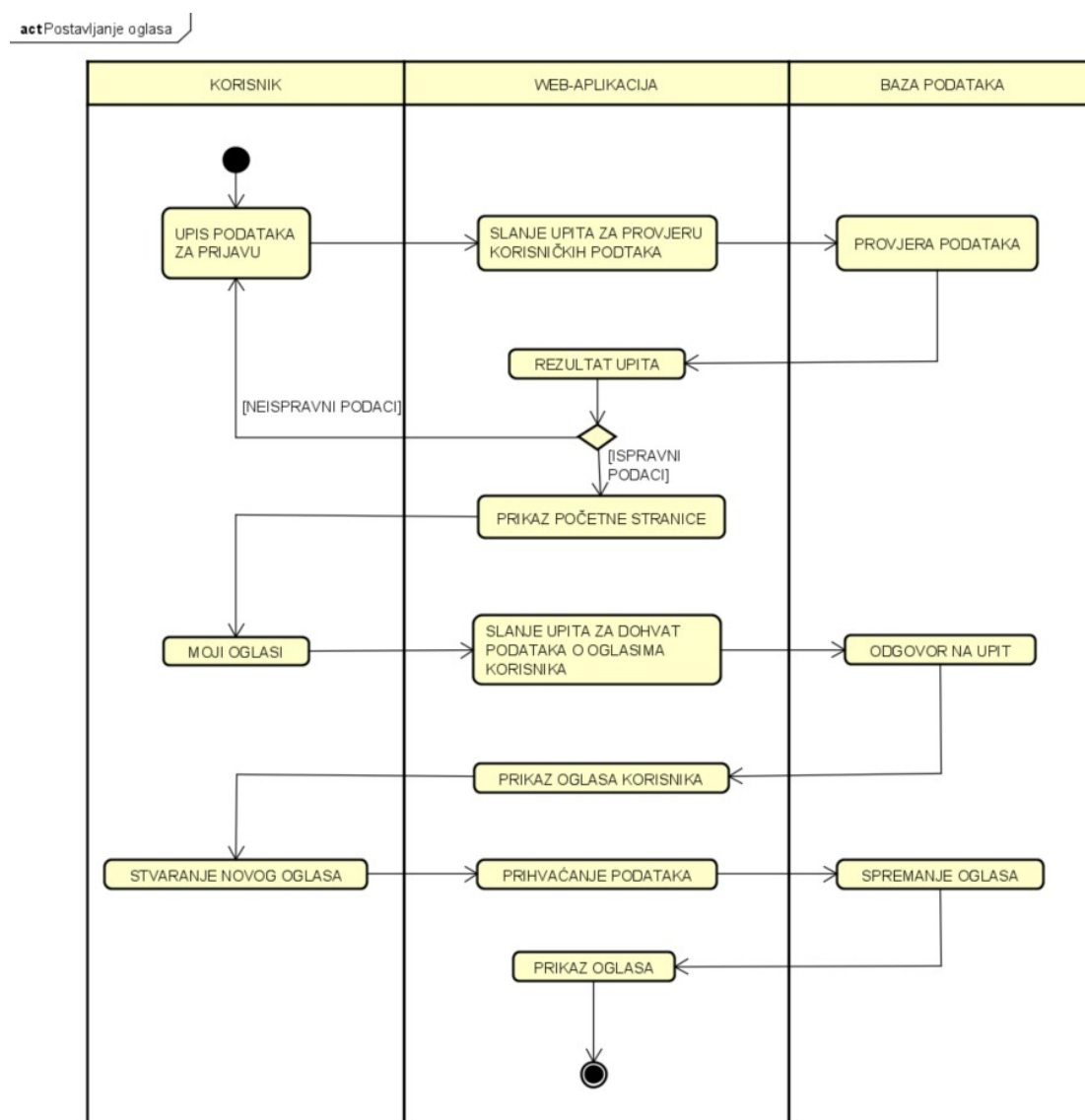




Slika 4.4: Dijagram stanja - registrirani korisnik

## 4.4 Dijagram aktivnosti

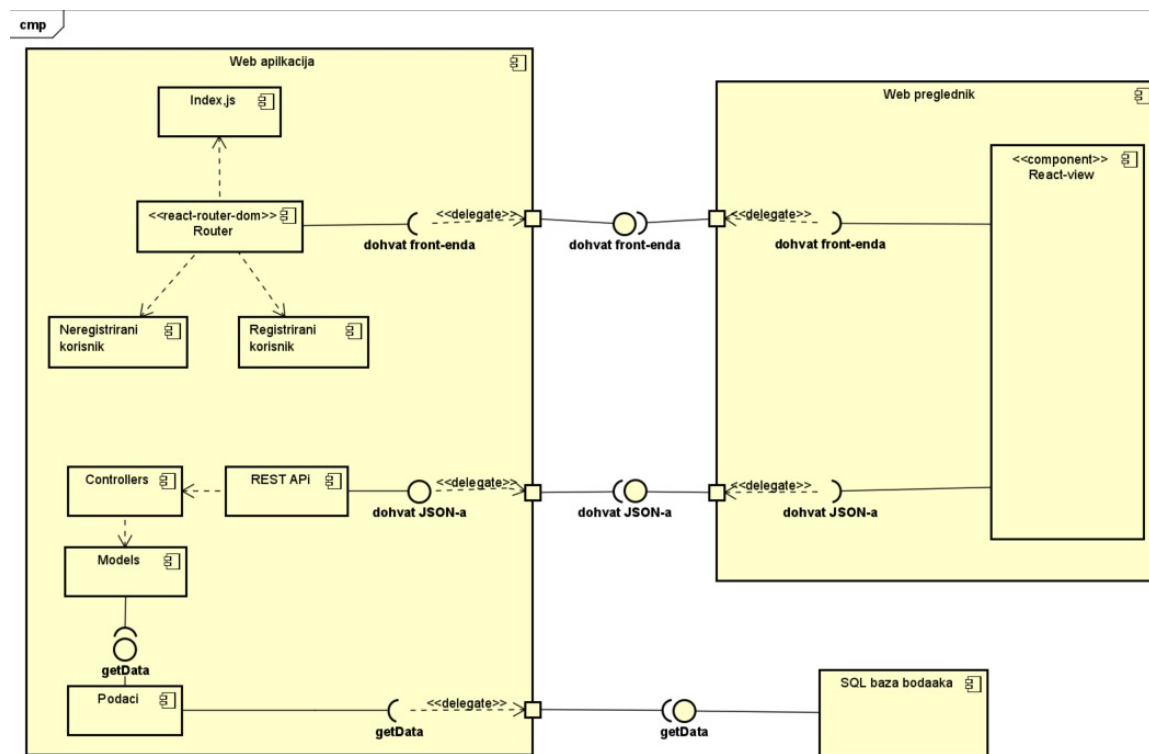
Dijagram aktivnosti je ponašajni UML-dijagram koji se koristi za modeliranje i vizualizaciju dinamičkog ponašanja sustava. Prikazuje izvođenje aktivnosti nizom akcija koje čine tokove upravljanja ili tokove podataka, s naglaskom na slijed i uvjete toka. Na slici 4.5 prikazan je dijagram aktivnosti za proces postavljanja oglasa. Korisnik se najprije prijavljuje u sustav, odabire svoje oglase, te opciju stvaranja novog oglasa. Korisnik ispunjava podatke o izgubljenom kućnom ljubimcu. Nakon što je korisnik popunio i spremio podatke, oglas postaje vidljiv i ostalim korisnicima.



Slika 4.5: Dijagram aktivnosti - postavljanje oglasa

## 4.5 Dijagram komponenti

Pristup sustavu ostvaruje se putem dva različita sučelja. Sučelje za dohvat front-end datoteka služi za obradu datoteka koju pripadaju front-end dijelu aplikacije. Router je komponenta koja na upit s URL-a određuje koja datoteka će se poslužiti na cilju. Frontend dio sastoji se od niza HTML, CSS i JavaScript datoteka. Sve front-end datoteke ovise o React biblioteci, iz koje dohvaćaju gotove komponente kao što su gumbi, forme i slično. Cilj za dohvat JSON podataka pristupa REST API komponenti, koja poslužuje podatke pripadajuće back-end dijelu aplikacije. Podaci se dohvaćaju iz baze podataka pomoću SQL upita. Podaci iz baze šalju se dalje MVC arhitekturi u obliku modela. Reactview komponenta, putem dostupnih sučelja, komunicira sa sjedištem aplikacije i ovisno o korisničkim akcijama osvježava prikaz te dohvaća nove podatke ili datoteke.



Slika 4.6: Dijagram komponenti

## 5. Implementacija i korisničko sučelje

### 5.1 Korištene tehnologije i alati

Komunikacija među članovima tima odvijala se putem aplikacije WhatsApp<sup>1</sup>. Za izradu UML dijagrama korišten je alat Astah Professional<sup>2</sup>. Za upravljanje izvornim kodom korišten je Git<sup>3</sup>, a kao udaljeni repozitorij projekta git platforma GitHub<sup>4</sup>.

Kao razvojno okruženje korišten je Microsoft Visual Studio<sup>5</sup>, koji pruža različite alate za razvoj računalnih programa kao što su web stranice, web usluge i mobilne aplikacije.

Za izradu backenda aplikacije korišten je radni okvir .NET Framework<sup>6</sup> i jezik C#<sup>7</sup>, a za izradu frontenda React<sup>8</sup> i jezik JavaScript<sup>9</sup>. .NET Framework je razvojni okvir tvrtke Microsoft koji omogućava razvoj, implementaciju i pokretanje web aplikacija. Pri razvoju aplikacija nudi gotova rješenja i funkcionalnosti kako bi se ubrzao i pojednostavio proces razvoja aplikacije. React je knjižnica pisana u JavaScriptu, a koristi se za izgradnju korisničkih sučelja ili njegovih dijelova.

---

<sup>1</sup><https://web.whatsapp.com/>

<sup>2</sup><https://astah.net/products/astah-uml/>

<sup>3</sup><https://git-scm.com/>

<sup>4</sup><https://github.com/>

<sup>5</sup><https://visualstudio.microsoft.com/>

<sup>6</sup><https://dotnet.microsoft.com/en-us/>

<sup>7</sup><https://learn.microsoft.com/en-us/dotnet/csharp/>

<sup>8</sup><https://react.dev/>

<sup>9</sup><https://www.javascript.com/>

## 5.2 Ispitivanje programskog rješenja

Ispitivanje je aktivnost s ciljem poboljšanja kvalitete proizvoda tako da se pronalaze i otklanjaju greške u implementaciji. Glavni pojam u ispitivanju je ispitni slučaj koji se definira kao uređeni par ulaznog podatka i očekivanog izlaznog podatka, koji je zabilježen prije provođenja ispitivanja.

### 5.2.1 Ispitivanje komponenti

Ispitivanje jedinica podrazumijeva ispitivanje funkcionalnosti najmanjih dijelova programa kao što su razredi, pripadajuće metode i atributi.

#### Ispitni slučaj 1.: Unos imena

Ulaz:

1. Korisnik unosi ispravno ime
2. Korisnik unosi ispravno ime koje sadrži diakritičke znakove
3. Korisnik unosi ime koje sadrži broj
4. Korisnik unosi ime koje sadrži nedopuštene simbole

Očekivani rezultat:

1. Provjera je uspješna i ime je valjano
2. Provjera je uspješna i ime je valjano
3. Sustav javlja da ime nije valjano
4. Sustav javlja da ime nije valjano

Rezultat:

Testovi su zadovoljeni, aplikacija je prošla test.

```
//testiramo firstName
List<string> firstNames = new List<string>();

// Add elements to the list

firstNames.Add("Valentino"); //ispravno
firstNames.Add("Šime"); // unos slova č,ć,ž,š itd., također ispravno
firstNames.Add("Lana34"); //unos broja
firstNames.Add("Lana€"); //unos simbola

foreach (string firstName in firstNames)
{
    IWebDriver driver = new ChromeDriver();
    RegistrationTest.performRegistration(driver, "valentino.boskovic@gmail.com", "0997421334", firstName, "Boskovic");
    driver.Quit();
}
```

Slika 5.1: Test ispravnosti imena kod registracije

### Ispitni slučaj 2.: Unos prezimena

#### Ulaz:

1. Korisnik unosi ispravno prezime
2. Korisnik unosi ime koje sadrži broj
3. Korisnik unosi ime koje sadrži nedopuštene simbole

#### Očekivani rezultat:

1. Provjera je uspješna i prezime je valjano
2. Sustav javlja da prezime nije valjano
3. Sustav javlja da prezime nije valjano

#### Rezultat:

Testovi su zadovoljeni, aplikacija je prošla test.

```
//testiramo lastName
List<string> lastNames = new List<string>();

// Add elements to the list
lastNames.Add("Bošković"); //ispravno
lastNames.Add("Horvat2"); // unos broja
lastNames.Add("Horvat$$"); //unos znaka

foreach (string lastName in lastNames)
{
    IWebDriver driver = new ChromeDriver();
    RegistrationTest.performRegistration(driver, "valentino.boskovic@gmail.com", "0997421334", "Valentino", lastName);
    driver.Quit();
}
```

Slika 5.2: Test ispravnosti prezimena kod registracije

### Ispitni slučaj 3.: Unos broja telefona

#### Ulaz:

1. Korisnik unosi ispravan broj mobitela
2. Korisnik unosi slova i/ili simbole
3. Korisnik unosi broj s crticama
4. Korisnik unosi broj s razmacima

#### Očekivani rezultat:

1. Provjera je uspješna i broj mobitela je valjan
2. Sustav javlja da broj mobitela nije valjan
3. Sustav javlja da broj mobitela nije valjan

4. Sustav javlja da broj mobitela nije valjan

**Rezultat:**

Testovi su zadovoljeni, aplikacija je prošla test.

```
//testiramo phoneNum
List<string> phoneNums = new List<string>();

// Add elements to the list
phoneNums.Add("0997421334"); //ispravno
phoneNums.Add("099a78954$"); // unos slova i/ili znaka
phoneNums.Add("099-742-1334"); //s crticama
phoneNums.Add("099 742 1334"); //razmaci

foreach (string phoneNum in phoneNums)
{
    IWebDriver driver = new ChromeDriver();
    RegistrationTest.performRegistration(driver, "valentino.boskovic@gmail.com", phoneNum, "Valentino", "Boskovic");
    driver.Quit();
}
```

Slika 5.3: Test ispravnosti broja mobitela kod registracije

**Ispitni slučaj 4.: Unos e-mail adrese****Ulaz:**

1. Korisnik unosi ispravnu e-mail adresu
2. Korisnik unosi e-mail adresu bez simbola @
3. Korisnik unosi samo domen
4. Korisnik unosi e-mail adresu bez domene
5. Korisnik unosi e-mail adresu bez ekstenzije

**Očekivani rezultat:**

1. Provjera je uspješna i e-mail adresa je valjana
2. Sustav javlja da e-mail adresa nije valjana
3. Sustav javlja da e-mail adresa nije valjana
4. Sustav javlja da e-mail adresa nije valjana
5. Sustav javlja da e-mail adresa nije valjana

**Rezultat:**

Testovi su zadovoljeni, aplikacija je prošla test.

```
//testiramo email
List<string> emails = new List<string>();

// Add elements to the list
emails.Add("valentino.boskovic@gmail.com"); //ispravno
emails.Add("ivanhorvat.com"); //bez @
emails.Add("@fer.com"); //bez usernamea samo domena
emails.Add("horvatNina@.com"); //bez domene
emails.Add("horvatNina@fer."); //bez ekstenzije

foreach (string email in emails)
{
    IWebDriver driver = new ChromeDriver();
    RegistrationTest.performRegistration(driver, email, "0992168933", "Valentino", "Boskovic");
    driver.Quit();
}
```

Slika 5.4: Test ispravnosti e-mail adrese kod registracije

### 5.2.2 Ispitivanje sustava

Ispitivanje sustava je proces ispitivanja završene i potpuno integrirane inačice programa namijenjene distribuciji korisniku.

#### Ispitni slučaj 1.: Login

Ulaz:

1. Korisnik unosi ispravno korisničko ime i lozinku
2. Korisnik unosi neispravno korisničko ime i lozinku

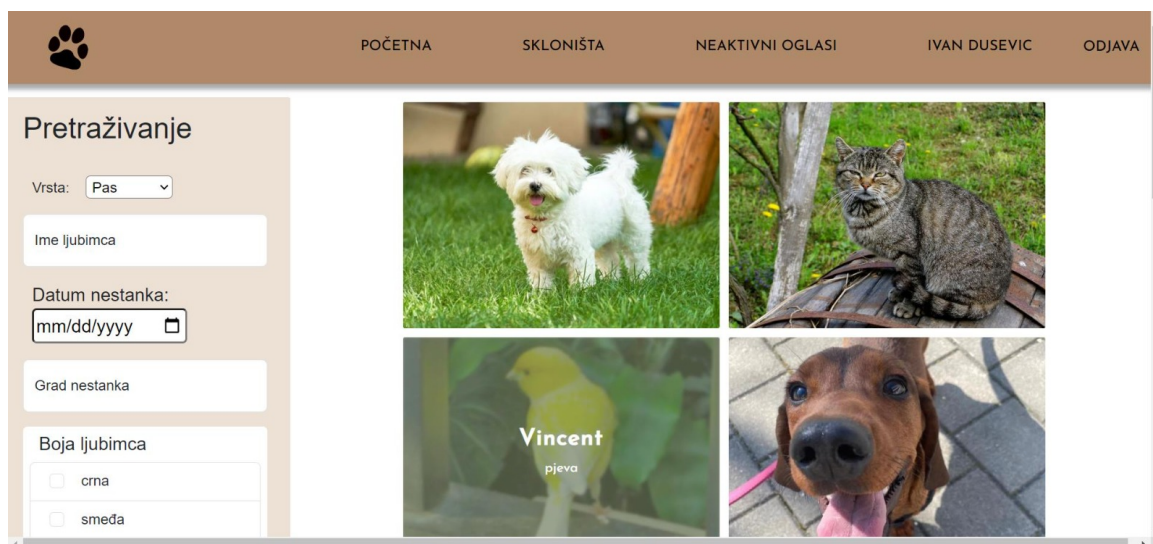
Očekivani rezultat:

1. Sustav obavještava korisnika da je uneseno korisničko ime neispravno ili lozinka neispravna
2. Prikazuje se početna stranica

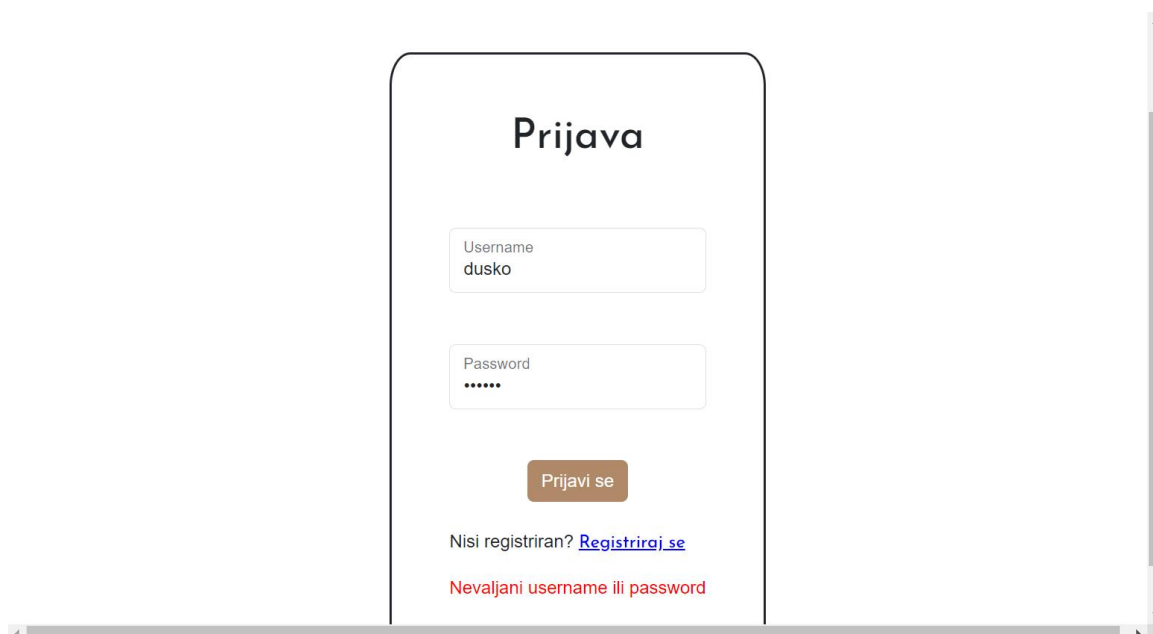
**Rezultat:**

Testovi su zadovoljeni, aplikacija je prošla test.





Slika 5.5: Ispravan login



Slika 5.6: Neispravan login

```
}  
public static void performLogin(IWebDriver driver, string username, string password)  
{  
    try  
    {  
        // Navigate to the login page  
        driver.Navigate().GoToUrl("https://localhost:5173/login");  
  
        // Find the username and password input fields and the login button  
        IWebElement usernameInput = driver.FindElement(By.Id("floatingInput"));  
        IWebElement passwordInput = driver.FindElement(By.Id("floatingPassword"));  
        IWebElement loginButton = driver.FindElement(By.Id("btn"));  
  
        usernameInput.SendKeys(username);  
        passwordInput.SendKeys(password);  
  
        // Click the login button  
        loginButton.Click();  
  
        WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));  
  
        wait.Until(driver => driver.Url.Equals("https://localhost:5173/"));  
  
        Thread.Sleep(3000); //wait so page loads correctly  
  
        string fileName = @"C:\Users\Korisnik\Desktop\selenium\SeleniumWebDriver_MP7\SeleniumWebDriver_MP7\screenshots\login_successful.png";  
        TakeScreenshot(driver, fileName);  
    }  
    catch (Exception ex)  
    {  
        string fileName = @"C:\Users\Korisnik\Desktop\selenium\SeleniumWebDriver_MP7\SeleniumWebDriver_MP7\screenshots\login_failed.png";  
        TakeScreenshot(driver, fileName);  
        Console.WriteLine(ex.Message);  
    }  
}
```

Slika 5.7: Test ispravnosti prijave u sustav

## Ispitni slučaj 2.: Registracija

### Ulaz:

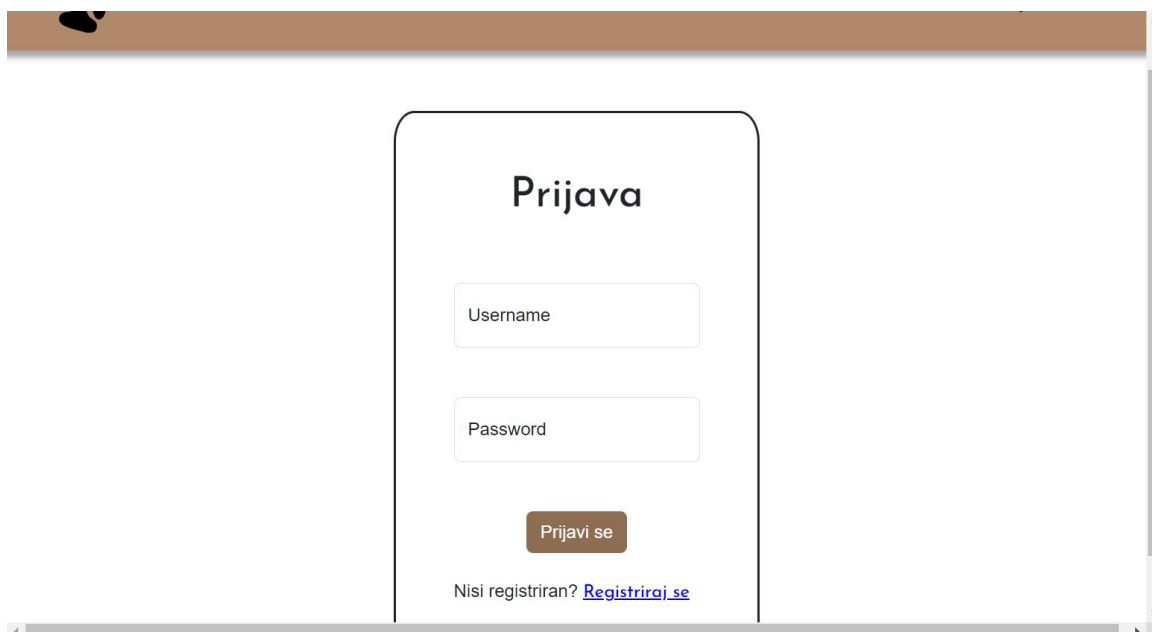
1. Korisnik unosi ispravno sve podatke

### Očekivani rezultat:

1. Sustav prikazuje stranicu za login

### Rezultat:

Testovi su zadovoljeni, aplikacija je prošla test.



Slika 5.8: Uspješna registracija

```
try
{
    driver.Navigate().GoToUrl("https://localhost:5173/registration");

    IWebElement usernameInput = driver.FindElement(By.Id("floatingInput"));
    IWebElement passwordInput = driver.FindElement(By.Id("floatingPassword"));
    IWebElement emailInput = driver.FindElement(By.Id("floatingEmail"));
    IWebElement phoneNumInput = driver.FindElement(By.Id("floatingPhoneNumber"));
    IWebElement typeOfUser = driver.FindElement(By.Id("typeOfUser"));
    SelectElement select = new SelectElement(typeOfUser);

    //ako registriamo sklonište samo By.Id("floatingShelterName") i zakomentiramo LastNameInput
    IWebElement firstNameInput = driver.FindElement(By.Id("floatingFirstName"));
    IWebElement LastNameInput = driver.FindElement(By.Id("floatingLastName"));

    IWebElement registrationButton = driver.FindElement(By.Id("btn"));

    usernameInput.SendKeys("valentino_b");
    passwordInput.SendKeys("blojskaRapsodija123");
    emailInput.SendKeys("valentino.bosk@gmail.com");
    phoneNumInput.SendKeys("0992168933");

    //ako zelimo sklonište - "Sklonište"
    select.SelectByText("Korisnik");

    firstNameInput.SendKeys("Valentino");
    LastNameInput.SendKeys("Boskovic");

    Thread.Sleep(5000);

    registrationButton.Click();

    WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
    wait.Until(driver => driver.Url.Equals("https://localhost:5173/login"));

    Thread.Sleep(3000); //wait so page loads correctly

    string fileName = @"C:\Users\Korisnik\Desktop\selenium\SeleniumWebDriver_MP7\SeleniumWebDriver_MP7\screenshots\registration_successful.png";
}
```

Slika 5.9: Test ispravnosti registracije u sustav

### Ispitni slučaj 3.: Filtriranje oglasa

Ulaz:

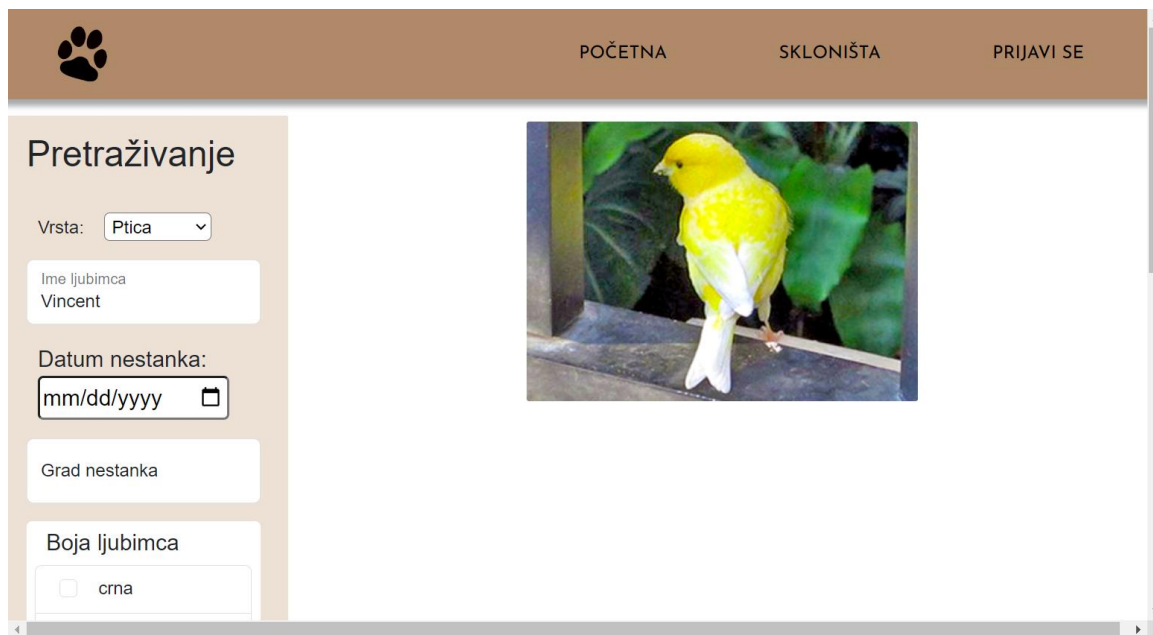
1. Korisnik odabire kategorije po kojima želi filtrirati oglase

**Očekivani rezultat:**

1. Sustav prikazuje filtrirane oglase

**Rezultat:**

Testovi su zadovoljeni, aplikacija je prošla test.



Slika 5.10: Uspješno filtriranje oglasa

```
public static void performSearch(IWebDriver driver)
{
    try
    {
        // Navigate to the login page
        driver.Navigate().GoToUrl("https://localhost:5173*");
        WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));

        // Find the username and password input fields and the login button
        IWebElement specie = driver.FindElement(By.Id("pet-species"));
        SelectElement selectSpecie = new SelectElement(specie);
        IWebElement nameInput = driver.FindElement(By.Id("pet-name"));
        IWebElement searchButton = driver.FindElement(By.Id("btn"));

        selectSpecie.SelectByText("Ptica");
        nameInput.SendKeys("Vincent");

        Thread.Sleep(5000);
        ((IJavaScriptExecutor)driver).ExecuteScript("arguments[0].scrollIntoView(true);", searchButton);
        searchButton.Click();

        Thread.Sleep(5000); //wait so page loads correctly

        string fileName = @"C:\Users\Korisnik\Desktop\selenium\SeleniumWebDriver_MP7\SeleniumWebDriver_MP7\screenshots\search_successful.png";
        TakeScreenshot(driver, fileName);
    }
    catch (Exception ex)
    {
        string fileName = @"C:\Users\Korisnik\Desktop\selenium\SeleniumWebDriver_MP7\SeleniumWebDriver_MP7\screenshots\search_failed.png";
        TakeScreenshot(driver, fileName);
        Console.WriteLine(ex.Message);
    }
}
```

Slika 5.11: Test ispravnosti filtriranja oglasa

#### Ispitni slučaj 4.: Objavljivanje oglasa

##### Ulaz:

1. Korisnik unosi tekst, sliku i geolokaciju
2. Korisnik ne unosi sliku
3. Korisnik ne unosi geolokaciju

##### Očekivani rezultat:

1. Stvara se oglas
2. Sustav traži da se unese barem jedna slika
3. Sustav traži da se unese geolokacija

##### Rezultat:

Testovi su zadovoljeni, aplikacija je prošla test.



Slika 5.12: Uspješno objavljivanje oglasa

```
public static void AdTest(IWebDriver driver)
{
    try
    {
        IWebElement myAdsButton = driver.FindElement(By.Id("btn-moji-oglas"));
        myAdsButton.Click();

        Thread.Sleep(3000);

        IWebElement newAddButton = driver.FindElement(By.Id("add-button"));
        newAddButton.Click();

        WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));

        wait.Until(driver => driver.Url.Equals("https://localhost:5173/newAd"));

        IWebElement fileInput = driver.FindElement(By.Id("img"));
        IWebElement imeInput = driver.FindElement(By.Id("ime"));
        IWebElement specie = driver.FindElement(By.Id("vrsta"));
        SelectElement selectSpecie = new SelectElement(specie);
        IWebElement checkbox = driver.FindElement(By.Name("crna-boja"));
        IWebElement dropdownAge = driver.FindElement(By.Id("age"));
        SelectElement selectAge = new SelectElement(dropdownAge);
        IWebElement dateInput = driver.FindElement(By.Id("datum-nestanka"));
        IWebElement timeInput = driver.FindElement(By.Id("vrijeme-nestanka"));
        IWebElement gradInput = driver.FindElement(By.Id("grad-nestanka"));
        IWebElement element = driver.FindElement(By.Id("map-container"));

        IWebElement descInput = driver.FindElement(By.Id("opis"));
        IWebElement btnSubmit = driver.FindElement(By.Id("btn-createAd"));

        string imgPath = @"C:\Users\Korisnik\Desktop\selenium\SeleniumWebDriver_MP7\SeleniumWebDriver_MP7\Tests\cat.jpeg";
        fileInput.SendKeys(imgPath);

        imeInput.SendKeys("Jura");
        selectSpecie.SelectByText("Mačka");

        IJavaScriptExecutor js = (IJavaScriptExecutor)driver;
```

Slika 5.13: Test ispravnosti objavljivanja oglasa



```
WebElement btnSubmit = driver.FindElement(By.Id("btn-createAd"));

string imgPath = @"C:\Users\Korisnik\Desktop\selenium\SeleniumWebDriver_MP7\SeleniumWebDriver_MP7\Tests\cat.jpeg";
fileInput.SendKeys(imgPath);

imeInput.SendKeys("Jura");
selectSpecie.SelectByText("Mačka");

IJavaScriptExecutor js = (IJavaScriptExecutor)driver;
js.ExecuteScript("arguments[0].click();", checkbox);
selectAge.SelectByText("6-10 god.");
dateInput.SendKeys("12-11-2023");
timeInput.SendKeys("14:30");
gradInput.SendKeys("Makarska");

descInput.SendKeys("jura juri");

((IJavaScriptExecutor)driver).ExecuteScript("arguments[0].click();", element);

Thread.Sleep(5000);

element.Click();

Thread.Sleep(5000);

((IJavaScriptExecutor)driver).ExecuteScript("arguments[0].scrollIntoView(true);", btnSubmit);
btnSubmit.Click();

Thread.Sleep(5000);

wait.Until(driver => driver.Url.Equals("https://localhost:5173/moji-oglas"));

string fileName = @"C:\Users\Korisnik\Desktop\selenium\SeleniumWebDriver_MP7\SeleniumWebDriver_MP7\screenshots\createAd_successful.png";
TakeScreenshot(driver, fileName);
}
catch (Exception ex)
{
    string fileName = @"C:\Users\Korisnik\Desktop\selenium\SeleniumWebDriver_MP7\SeleniumWebDriver_MP7\screenshots\createAd_failed.png";
    TakeScreenshot(driver, fileName);
    Console.WriteLine(ex.Message);
}
```

Slika 5.14: Test ispravnosti objavljivanja oglasa

```
//testiramo ne unešenu sliku pri kreiranju oglasa

Boolean unesiSliku = false;

IWebDriver driver = new ChromeDriver();
LoginTest.performLogin(driver, "danci", "kroksice");
PostAdTest.AdTest(driver, unesiSliku, true);
driver.Quit();

//testiramo ne unešenu lokaciju pri kreiranju oglasa

Boolean unesiLokaciju = false;
LoginTest.performLogin(driver, "danci", "kroksice");
PostAdTest.AdTest(driver, true, unesiLokaciju);
driver.Quit();
```

Slika 5.15: Test ispravnosti objavljivanja oglasa

### **Ispitni slučaj 5.: Komentiranje oglasa**

#### **Ulaz:**

1. Neregistrirani/neprijavljeni korisnik klikne na gumb "Dodaj komentar"

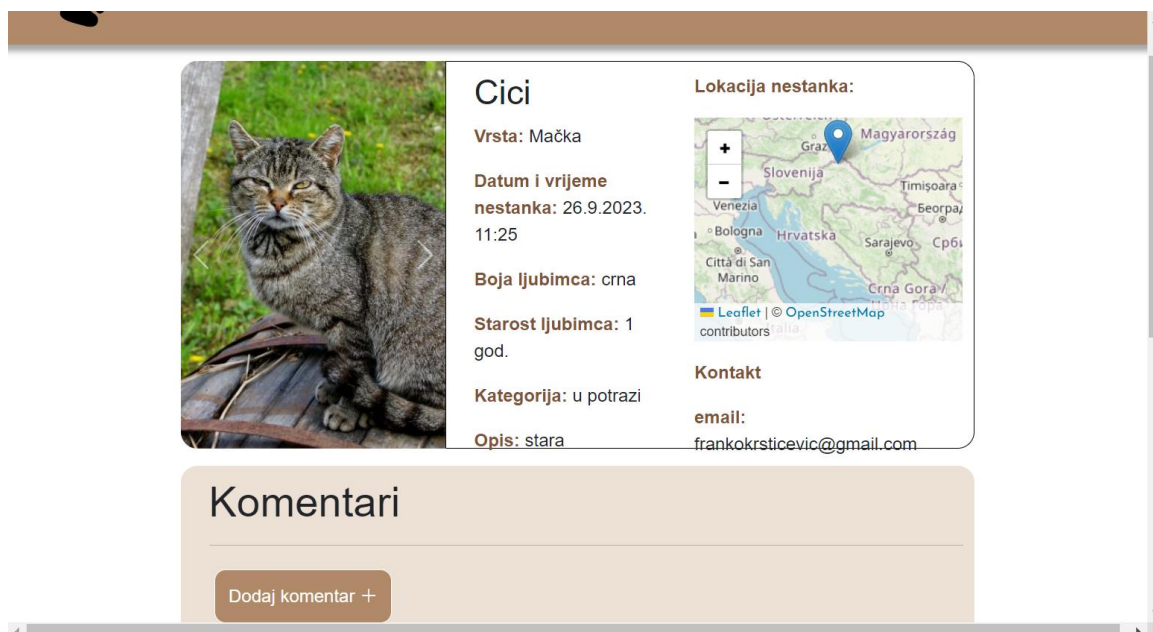
#### **Očekivani rezultat:**

1. Sustav ne dopušta korisniku da ostavi komentar

#### **Rezultat:**

Testovi su zadovoljeni, aplikacija je prošla test.





Slika 5.16: Komentiranje oglasa bez prijave

```
public static void CommentTest(IWebDriver driver)
{
    try
    {
        driver.Navigate().GoToUrl("https://localhost:5173/21");

        WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
        wait.Until(driver => driver.Url.Equals("https://localhost:5173/21"));

        Thread.Sleep(3000);

        IWebElement addComInput = driver.FindElement(By.Id("add-button"));
        addComInput.Click();

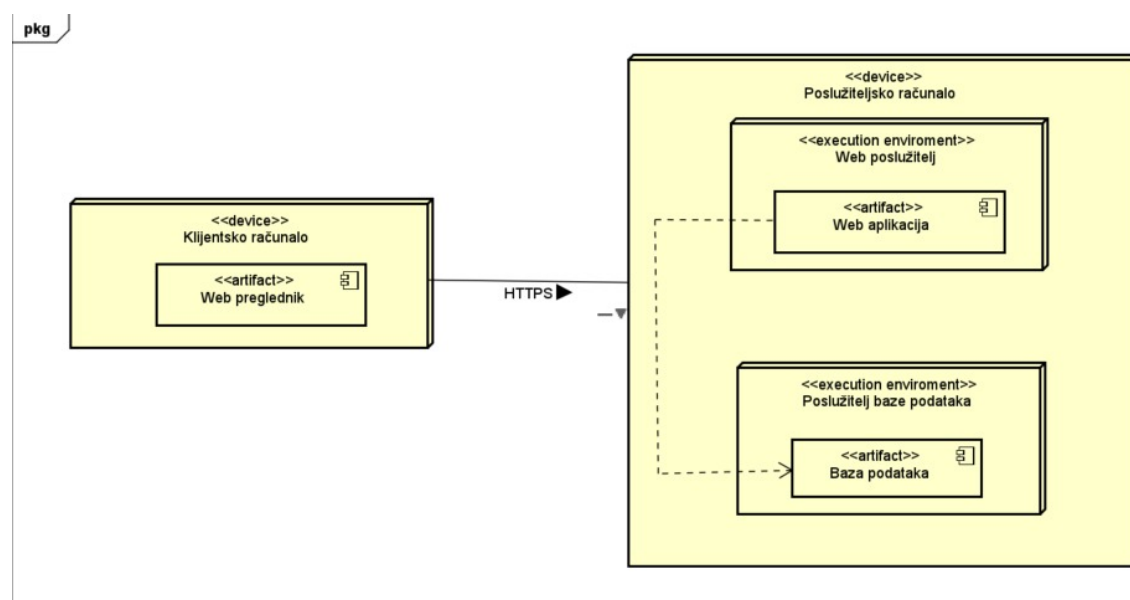
        IWebElement textareaInput = driver.FindElement(By.Id("floatingTextarea2"));
        IWebElement mapInput = driver.FindElement(By.ClassName("map-container"));
        IWebElement imgInput = driver.FindElement(By.ClassName("img-container"));
        IWebElement btnSubmit = driver.FindElement(By.TagName("submit"));

        textareaInput.SendKeys("This is my comment.");
    }
}
```

Slika 5.17: Test ispravnosti komentiranja oglasa

## 5.3 Dijagram razmještaja

Na poslužiteljskom računalu smješteni su web poslužitelj i poslužitelj baze podataka. Klijenti pristupaju web aplikaciji putem web preglednika. Sustav je temeljen na arhitekturi "klijent-poslužitelj", a komunikacija između računala korisnika i poslužitelja odvija se putem HTTP veze.



Slika 5.18: Dijagram razmještaja

## 5.4 Upute za puštanje u pogon

### 5.4.1 Konfiguracija baze podataka

Pošto smo koristili SQLite bazu podataka, nema posebnog dizanja servera baze. SQLite je baza realizirana pomoću file sistema. Bazu smo već napunjenu s podacima kopirali pomoću Dockerfile-a.

### 5.4.2 Izgradnja aplikacije

U webapi folderu smo pokrenuli naredbu „dotnet publish -c Release -o out“, time smo izgradili backend i kreirali out folder. Zatim smo se pozicionirali u reactapp folder i tamo izveli naredbu „npm run build“. Time smo dobili izgrađeni frontend aplikacije. Nakon toga smo iz foldera dist koji se napravio naredbom „npm run build“ kopirali file-ove u folder wwwroot unutar out foldera koji se nalazi unutar webapi foldera.

Pomoću dockera smo napisali naredbe koje će se izvršiti kod puštanja aplikacije u pogon na stranici render.com.

```
FROM mcr.microsoft.com/dotnet/sdk:7.0 AS build-env
WORKDIR /App

# Copy everything
COPY out/ ./
COPY MP7.db ./

# Restore as distinct layers
#RUN dotnet restore webapi
#RUN dotnet restore reactapp

# Build and publish a release
#RUN dotnet publish webapi -c Release -o out

# Build runtime image
FROM mcr.microsoft.com/dotnet/aspnet:7.0
WORKDIR /App

EXPOSE 5000/tcp

ENV ASPNETCORE_URLS http://*:5000

COPY --from=build-env /App .
ENTRYPOINT ["dotnet", "webapi.dll"]
```

Slika 5.19: DockerFile

Unutar Dockerfile-a smo kreirali radni folder. Naredbom „COPY out/ ./“ smo rekli da se kopira cijeli out folder u trenutni App folder, zatim smo naredbom „COPY MP7.db ./“ kopirali već napunjenu bazu podataka u App folder. Pokretanjem aplikacije pokrenut će se App folder i sve što se u njemu nalazi, a tamo smo sve kopirali.

Puno bolji način pokretanje aplikacije u pogon je da se sve odvija unutar Dockerfile-a, to uključuje izvršavanje naredbe „dotnet publish -c Release -o out“ i naredbe „npm run build“, ali imali smo nekih poteškoća oko toga pa smo lokalno izgradili aplikaciju i takvu ju samo kopirali.

### 5.4.3 Deploy

Deploy se radio preko stranice render.com. Na toj stranici smo povezali Github repozitorij preko kojeg se stranica pušta u pogon. Odabrali smo opciju New Web Service i ispunili potrebne podatke. Odabrali smo repozitorij s kojeg će se pustiti aplikacija u pogon, izabrali granu koja će se pustiti u pogon i napisali ime stranice. Također smo izabrali opciju Dockerfile za Runtime web servisa i postavili root directory. Render.com ne podržava .NET zato smo morali koristiti Dockerfile. Nakon toga smo pustili aplikaciju u pogon.

<b>Name</b> A unique name for your web service.	<input type="text" value="Missing pets"/>
<b>Region</b> The <a href="#">region</a> where your web service runs. Services must be in the same region to communicate privately and you currently have services running in <b>Frankfurt</b> .	<input type="text" value="Frankfurt (EU Central)"/>
<b>Branch</b> The repository branch used for your web service.	<input type="text" value="deploy"/>
<b>Root Directory</b> <small>Optional</small> Defaults to repository root. When you specify a <a href="#">root directory</a> that is different from your repository root, Render runs all your commands in the <a href="#">specified directory</a> and ignores changes outside the directory.	<input type="text" value="webapi"/>
<b>Runtime</b> The runtime for your web service.	<input type="text" value="Docker"/>

## 6. Zaključak i budući rad

Zadatak grupe bio je razvoj aplikacije za pomoć u potrazi izgubljenih kućnih ljubimaca uz mogućnost postavljanja oglasa o nestaloj životinji, pregled oglasa i komentiranje oglasa. Aplikacija je napravljena u 14 tjedana, koji su podijeljeni u dvije faze.

U prvoj fazi okupljen je tim za razvoj aplikacije od 7 članova i dodijeljen projektni zadatak. U ovoj fazi najintenzivniji rad je proveden na dokumentiranju zahtjeva, koje je bilo osnova za što lakšu implementaciju osmišljenog sustava. Specifikacija programske potpore, koja uključuje funkcionalne zahtjeve, obrasce uporabe i različite vrste UML dijagrama (dijagram obrazaca uporabe, sekvencijski dijagrami, dijagrami razreda, model baze podataka) služili su kao nacrti i orijentir za izradu frontenda i backenda.

Druga faza projekta bila je intenzivnija po pitanju implementacije. Kod implementacije programskog rješenja nije bilo većih problema zbog spremnosti članova grupe na pomoć i samostalno učenje. U ovoj fazi izrađeni su preostali dijagrami (dijagram stanja, aktivnosti, komponenti i razmještaja) te ostatak dokumentacije.

Sudjelovanje na ovom projektu bilo je vrijedno iskustvo svim članovima grupe jer nas se većina po prvi put susrela s ovakvom vrstom zadatka u kojem je bila potrebna visoka razina organizacije, spremnost za rad u timu, samostalno učenje, disciplina i redovit rad. Zaključno, članovi tima su zadovoljni izrađenom aplikacijom te su implementirane sve planirane funkcionalnosti.

# Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. Procesi programskog inženjerstva, [https://www.fer.unizg.hr/\\_download/repository/Procesi%20programskog%20inzenjerstva\\_5\\_izdanje.pdf](https://www.fer.unizg.hr/_download/repository/Procesi%20programskog%20inzenjerstva_5_izdanje.pdf)
3. The Unified Modeling Language, <https://www.uml-diagrams.org/>

# Indeks slika i dijagrama

2.1	Registracija . . . . .	8
2.2	Prijava . . . . .	8
2.3	Primjer oglasa . . . . .	10
3.1	Dijagram obrasca uporabe, funkcionalnosti neregistriranog i registriranog korisnika . . . . .	19
3.2	Sekvencijski dijagram za UC3 . . . . .	20
3.3	Sekvencijski dijagram za UC7 . . . . .	21
3.4	Sekvencijski dijagram za UC12 . . . . .	22
4.1	ER dijagram baze podataka . . . . .	30
4.2	Dijagram razreda - dio Controllers . . . . .	30
4.3	Dijagram razreda - dio Models . . . . .	31
4.4	Dijagram stanja - registrirani korisnik . . . . .	32
4.5	Dijagram aktivnosti - postavljanje oglasa . . . . .	33
4.6	Dijagram komponenti . . . . .	34
5.1	Test ispravnosti imena kod registracije . . . . .	36
5.2	Test ispravnosti prezimena kod registracije . . . . .	37
5.3	Test ispravnosti broja mobitela kod registracije . . . . .	38
5.4	Test ispravnosti e-mail adrese kod registracije . . . . .	39
5.5	Ispravni login . . . . .	40
5.6	Neispravni login . . . . .	40
5.7	Test ispravnosti prijave u sustav . . . . .	41
5.8	Uspješna registracija . . . . .	42
5.9	Test ispravnosti registracije u sustav . . . . .	42
5.10	Uspješno filtriranje oglasa . . . . .	43
5.11	Test ispravnosti filtriranja oglasa . . . . .	44
5.12	Uspješno objavljivanje oglasa . . . . .	45
5.13	Test ispravnosti objavljivanja oglasa . . . . .	45
5.14	Test ispravnosti objavljivanja oglasa . . . . .	46



5.15 Test ispravnosti objavljivanja oglasa . . . . .	47
5.16 Komentiranje oglasa bez prijave . . . . .	48
5.17 Test ispravnosti komentiranja oglasa . . . . .	48
5.18 Dijagram razmještaja . . . . .	49
5.19 DockerFile . . . . .	51
6.1 Prikaz aktivnosti na repozitoriju . . . . .	62

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### *Kontinuirano osvježavanje*

#### 1. sastanak

- Datum: 18. listopada 2023.
- Prisustvovali: Mia Krstičević, Filip Smolić, Toni Vanjak, Fran Kufrin, Dominik Šarić, Vito Tomas, Lucija Renić
- Teme sastanka:
  - općenito o zadatku
  - alati i tehnologije

#### 2. sastanak

- Datum: 23. listopada 2023.
- Prisustvovali: Mia Krstičević, Filip Smolić, Toni Vanjak, Fran Kufrin, Dominik Šarić, Vito Tomas, Lucija Renić
- Teme sastanka:
  - funkcionalnosti
  - zahtjevi
  - alati i tehnologije

#### 3. sastanak

- Datum: 2. studenoga 2023.
- Prisustvovali: Mia Krstičević, Filip Smolić, Toni Vanjak, Fran Kufrin, Dominik Šarić, Vito Tomas, Lucija Renić
- Teme sastanka:
  - alati i tehnologije
  - baza podataka

#### 4. sastanak

- Datum: 9. studenoga 2023.

- Prisustvovali: Mia Krstičević, Filip Smolić, Toni Vanjak, Fran Kufrin, Dominik Šarić, Vito Tomas, Lucija Renić
- Teme sastanka:
  - dosadašnji napredak
  - arhitektura

#### 5. sastanak

- Datum: 14. studenoga 2023.
- Prisustvovali: Mia Krstičević, Filip Smolić, Toni Vanjak, Fran Kufrin, Dominik Šarić, Vito Tomas, Lucija Renić
- Teme sastanka:
  - dosadašnji napredak
  - dijagram razreda

#### 6. sastanak

- Datum: 6. prosinca 2023.
- Prisustvovali: Mia Krstičević, Filip Smolić, Toni Vanjak, Fran Kufrin, Dominik Šarić, Vito Tomas, Lucija Renić
- Teme sastanka:
  - preraspodjela posla

#### 7. sastanak

- Datum: 13. prosinca 2023.
- Prisustvovali: Mia Krstičević, Filip Smolić, Toni Vanjak, Fran Kufrin, Dominik Šarić, Vito Tomas, Lucija Renić
- Teme sastanka:
  - dosadašnji napredak
  - dijagram stanja

#### 8. sastanak

- Datum: 10. siječnja 2024.
- Prisustvovali: Mia Krstičević, Filip Smolić, Toni Vanjak, Fran Kufrin, Dominik Šarić, Vito Tomas, Lucija Renić
- Teme sastanka:
  - dosadašnji napredak
  - dijagram razreda

#### 9. sastanak

- Datum: 16. siječnja 2024.

- Prisustvovali: Mia Krstičević, Filip Smolić, Toni Vanjak, Fran Kufrin, Dominik Šarić, Vito Tomas, Lucija Renić
- Teme sastanka:
  - dovršavanje implementacije
  - ispitivanje programskog rješenja

## Tablica aktivnosti

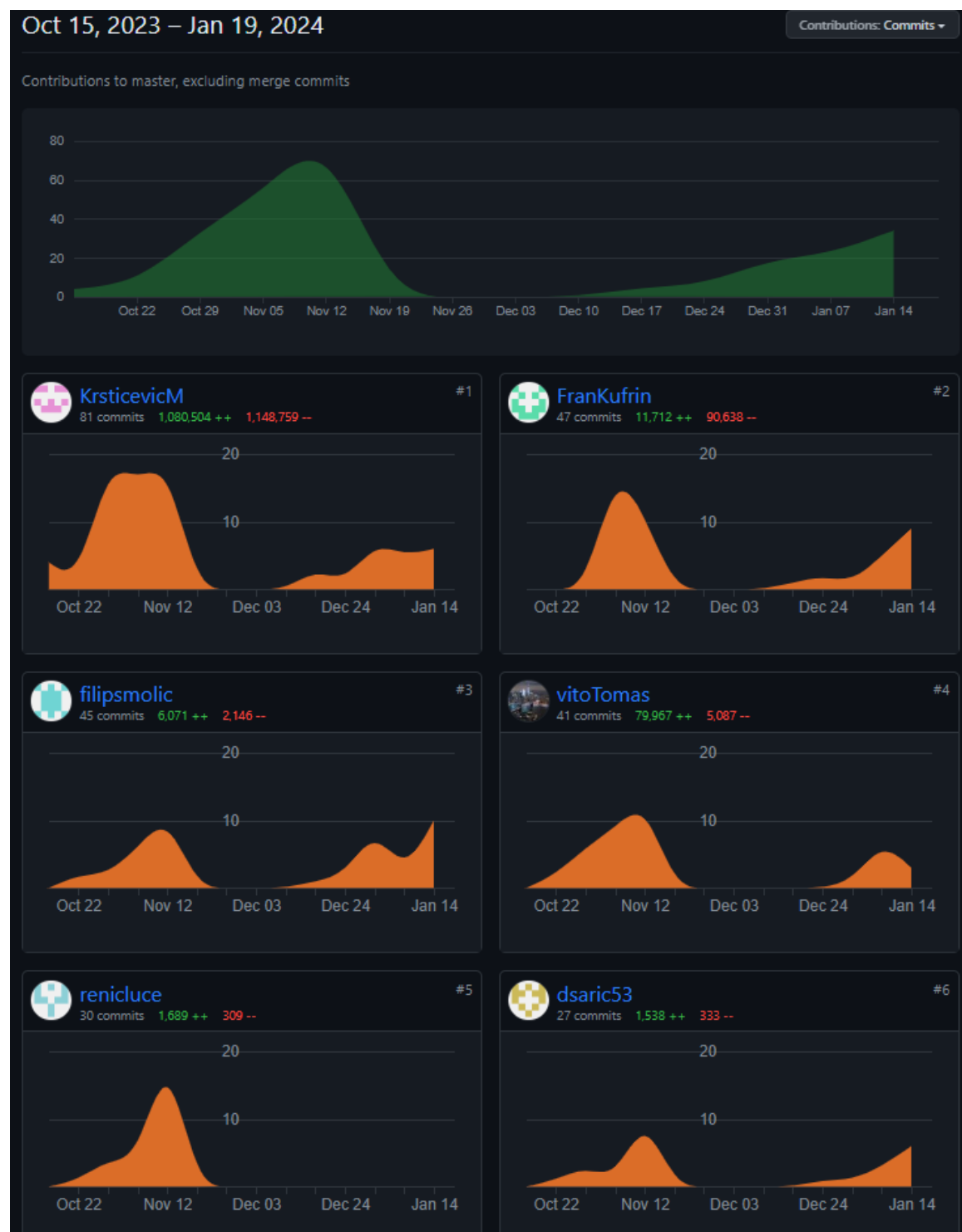
	Mia Krstičević	Filip Smolić	Toni Vanjak	Fran Kufrin	Dominik Šarić	Vito Tomas	Lucija Renić
Upravljanje projektom	40						
Opis projektnog zadatka							8
Funkcionalni zahtjevi							4
Opis pojedinih obrazaca							12
Dijagram obrazaca			5				3
Sekvencijski dijagrami			7				1
Opis ostalih zahtjeva							3
Arhitektura i dizajn sustava			6				5
Baza podataka	10						
Dijagram razreda			6				1
Dijagram stanja			1				3
Dijagram aktivnosti			1				3
Dijagram komponenti			3				1
Korištene tehnologije i alati							3
Ispitivanje programskog rješenja	18						15
Dijagram razmještaja			3				1
Upute za puštanje u pogon							
Dnevnik sastajanja							3
Zaključak i budući rad							3
Popis literature							1

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Mia Krstičević	Filip Smolić	Toni Vanjak	Fran Kufrin	Dominik Šarić	Vito Tomas	Lucija Renić
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>							
<i>front end</i>	2	170		140			
<i>izrada baze podataka</i>	63						
<i>spajanje s bazom podataka</i>					27	30	
<i>back end</i>	65				138	150	

## Dijagrami pregleda promjena



Slika 6.1: Prikaz aktivnosti na repozitoriju